VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

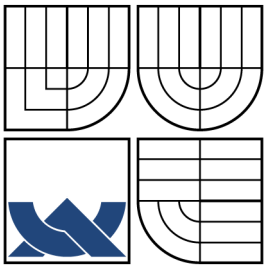
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

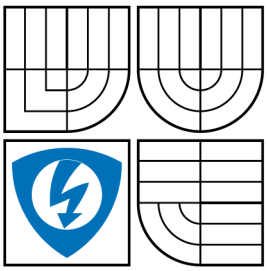
AUTENTIZACE POMOCÍ MOBILNÍHO TELEFONU

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE VILIAM KRIŽAN
AUTHOR



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

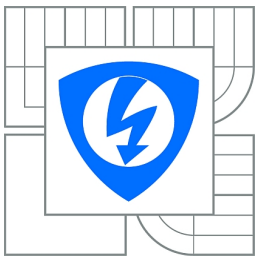
AUTENTIZACE POMOCÍ MOBILNÍHO TELEFONU
AUTHENTICATION USING A MOBILE PHONE

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
VILIAM KRIŽAN

VEDÚCI PRÁCE
SUPERVISOR
Ing. LUKÁŠ MALINA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Viliam Križan ***ID:*** 134529
Ročník: 3 ***Akademický rok:*** 2012/2013

NÁZEV TÉMATU:

Autentizace pomocí mobilního telefonu

POKYNY PRO VYPRACOVÁNÍ:

Zaměřte se na využití autentizace pomocí mobilních telefonů v přístupových systémech. Analyzujte a zhodnoťte možnosti autentizačních metod na soudobých mobilních telefonech s platformou android. Zaměřte se také na autentizaci uživatele vůči určité aplikaci na mobilním telefonu.
Cílem bakalářské práce bude návrh a implementace autentizačního protokolu využívající mobilní telefon. Komunikaci mobilního telefonu se čtecím zařízením zajistěte pomocí přenosové technologie NFC, popřípadě využijte jinou přenosovou technologii. Implementaci otestujte a zhodnoťte bezpečnost daného řešení a možnosti využití bezpečného uložení dat v telefonu pomocí tzv. secure elementu.

DOPORUČENÁ LITERATURA:

[1] STALLINGS, William. Cryptography and Network Security. 4th edition. [s.l.] : [s.n.], 2006. 592 s. ISBN 0131873164.
[2] MENEZES, Alfred, VAN OORSCHOT, Paul, VANSTONE, Scott. Handbook of applied cryptography. Boca Raton: CRC Press, 1997. 780 s. ISBN 0849385237.
[3] UJBÁNYAI, Miroslav. Programujeme pro Android. Vyd. 1. Praha: Grada, 2012, 187 s. ISBN 978-80-247-3995-3.
[4] MURPHY, Mark L. Android 2: průvodce programováním mobilních aplikací. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.

Termín zadání: 11.2.2013 ***Termín odevzdání:*** 5.6.2013

Vedoucí práce: Ing. Lukáš Malina
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

ABSTRAKT

Bakalárska práca popisuje základné autentizačné metódy znalosťou, používané autentizačné predmety a biometrické spôsoby autentizácie. Rozoberá možnosti aplikácie jednotlivých metód určených pre mobilné telefóny. Zaoberá sa základným zabezpečením v sieti GSM (s použitím SIM karty) a používaním autentizácie pomocou hesla, PIN kódu, gesta, reči a tváre. Jednotlivé metódy zhodnocuje z hľadiska úspešnosti a náročnosti. Ďalej rozoberá možnosti použitia autentizačných metód na platforme Android, kde uvádza operácie modulárnej aritmetiky kryptografických primitívov. Časovú náročnosť operácií modulárnej aritmetiky vyhodnocuje na mobilných telefónoch s pomocou vytvorenej aplikácie. Taktiež vymenúva dostupné algoritmy platformy Android: symetrické a asymetrické šifry, jednocestné hashe. V návrhu autentizačného protokolu rozpisuje protokol založený na dôkaze nulovej znalosti, ktorý užívateľov autentizuje anonymne. Navrhuje taktiež prenos vypočítaných autentizačných premenných pomocou technológie NFC. Protokol podporuje aj revokáciu nechcených užívateľov. Cieľom práce bolo vytvoriť dve aplikácie, ktoré majú implementovaný zvolený autentizačný protokol. Prvá aplikácia, určená pre užívateľa, sa spúšťa na mobilných telefónoch s operačným systémom Android. Aplikácia vypočítava autentizačné premenné z paramterov, ktoré si načítava z tzv. secure elementu telefónu. Premenné po vypočítaní posiela pomocou NFC čítaciemu zariadeniu (spojenie telefón – čítacie zariadenie). Čítacie zariadenie je pripojené k druhej aplikácií, ktorá užívateľove autentizačné premenné overuje a rozhoduje o úspešnosti autentizácie. Práca sa taktiež venuje testovaniu vytvorených aplikácií a ich spustenia na zariadeniach. Zhodnocuje bezpečnosť zvoleného autentizačného protokolu, využitie secure elementu telefónu a bezpečnosť prenosu pomocou NFC.

KĽÚČOVÉ SLOVÁ

Android, anonymita, autentizácia, Java, mobilný telefón, NFC, poverovacie tokeny

ABSTRACT

Bachelor's thesis describes the basics of authentication methods by knowledge, used authentication tokens and biometric ways of authentication. It discusses the capabilities of the individual methods designed for mobile phones. It covers the basic security in the GSM network (using SIM cards) and usage of authentication based on passwords, PIN codes, gestures, facial recognition and speech recognition. Thesis evaluates individual methods in terms of success and performance. It discusses the usage possibilities of authentication methods on the Android platform, where it provides the options of modular arithmetic operations of cryptographic primitives. The work evaluates the time consumption of modular arithmetic operations on the mobile phones with created application. It also lists available algorithms of symmetric and asymmetric ciphers, one-way hashes on the Android platform. The section of authentication protocol conception writes about protocol based on zero-knowledge proof, which authenticates users anonymously. It also proposes the transfer of calculated authentication variables using NFC technology. The protocol also supports the revocation of malicious users. The goal of the thesis was to create two applications that have implemented the selected authentication protocol. The first application designed for users, runs on mobile phones with Android operating system. Application calculates the authentication variables using authentication parameter which it loads from secure element of the phone. After calculations it sends the authentication variables via NFC reader (connection phone – reader). NFC reader is connected to the second application, which verifies the user's authentication variables and decides the conclusion of authentication. The last part of thesis is devoted to testing the developed applications and running them on the devices. It evaluates the security of the authentication protocol, usage of the phone's secure element and a security of the transmission over NFC.

KEYWORDS

Android, Anonymity, Authentication, Credentials, Mobile phone, NFC, Java

KRIŽAN, Viliam *Autentizace pomocí mobilního telefonu*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 58 s. Vedúci práce bol Ing. Lukáš Malina

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Autentizace pomocí mobilního telefonu“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno ....

(podpis autora)

POĎAKOVANIE

Rád by som sa poďakoval vedúcemu bakalárskej práce pánovi Ing. Lukášovi Malinovi za odborné vedenie, konzultácie a za cenné návrhy.

Brno ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

(podpis autora)

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ÚVOD

V praxi sa stretávame s autentizáciou na každom kroku, napr. pri ceste vlakom predkladáme cestovný lístok, ktorý značí úhradu cestovného. Pri vstupoch do univerzitných priestorov a laboratórií používame čipové karty.

Mobilné telefóny nosíme so sebou a využívame ich denne. Výkonné telefóny obsahujúce množstvo funkcií a taktiež mnoho senzorov označujeme ako chytré telefóny (tzv. smartphone-y). Môžeme ich použiť aj ako autentizačné predmety, namiesto čipových kariet alebo lístkov MHD. Chytré telefóny bežia na plnohodnotných operačných systémoch, akým je aj operačný systém Android. Pre ich masové využívanie sa stávajú terčom rôznych útokov, či už pri fyzickom odcudzení alebo napadnutí z internetu. Preto musíme chrániť ich obsah a funkcionalitu, aby niekto cudzí nezneužil našu identitu vo svoj prospech, pričom na ochranu môžeme využiť viacero metód autentizácie.

V kapitole 1 popisujeme základné metódy autentizácií, ktoré sa delia na autentizáciu znalosťou, autentizáciu predmetom a autentizáciu biometrikou. Použitie autentizačných metód a tzv. secure elementov na mobilných telefónoch rozoberá kapitola 2. Pojednáva aj o autentizácii v mobilnej sieti a využití SIM karty, základných autentizáciách užívateľ–telefón, akými sú autentizácia heslom, gestom, hlasom a tvárou. Vybrané autentizačné metódy subjektívne zhodnocuje podľa bezpečnosti, úspešnosti, náročnosti a potrebného vybavenia.

Pre prácu bol určený operačný systém Android. Stavba aplikácií a komponentov tohto systému je rozpísaná v kapitole 3, v ktorej sa taktiež venujeme možnostiam implementácie autentizačných schém. V kapitole sú taktiež popísané použiteľné symetrické alebo asymetrické šifry, jednocestné hashovacie funkcie, ale aj základné operácie modulárnej aritmetiky kryptografických primitívov, ktorých časovú náročnosť overujeme.

O návrhu autentizačného protokolu, ktorý využíva spojenia mobilný telefón – čítacie zariadenie, popisuje kapitola 4. Autentizačný protokol bol zvolený typom dôkazu nulovej znalosti, takže užívatelia pri autentizácii vystupujú anonymne. Protokol podporuje aj odoprenie určených užívateľov, spojiteľnosť s minulosťou, prípadne aj de-anonymizáciu v kritických udalostiach. V kapitole je rozpísaná dokazovacia fáza protokolu, použitie autentizačných parametrov a ich prenos. Taktiež pojednáva o komunikačnom prenose pomocou technológie NFC, enkapsulácii záznamov do správy a ich linkového zabezpečenia.

Implementáciou autentizačného protokolu sa zaoberá kapitola 5. Pojednáva o klientskej Android aplikácii, ktorá využíva modulárnu aritmetiku kryptografických primitívov pre výpočet autentizačných parametrov a pre ich prenos využíva NFC API systému Android. Taktiež popisuje serverovú aplikáciu, spustenú na osobnom počítači s pripojenou NFC čítačkou. Serverová aplikácia vystupuje v úlohe overovateľa (overuje užívateľov dôkaz) a využíva čiernu listinu pre vylúčených užívateľov. O testovaní a zhodnotení bezpečnosti navrhnutého protokolu diskutuje kapitola 6. Popisuje bezpečnostné nedostatky protokolu, využitie secure elementu v klientskej aplikácii a jej zabezpečenia, zhodnocuje taktiež bezpečnosť NFC komunikácie.

1 AUTENTIZAČNÉ METÓDY

Autentizácia je proces overenia identity **žiadateľa** (entity), ktorý vyžaduje prístup ku chráneným **aktívam** (budovy, dáta, služby). Žiadateľ je overený osobou alebo zariadením (**kontrolérom**) na základe predloženia svojich identifikačných údajov, ktoré sú pre daného žiadateľa jedinečné. O hĺbke a právach prístupu k aktívam rozhoduje osoba alebo orgán nazývaný **autorita**. Tento proces rozhodnutia označujeme ako **autorizácia**, ktorá má za účel chrániť aktíva pred nepovolanými entitami.

Autentizáciu môžeme rozdeliť do troch základných tried:

a) **autentizácia znalosťou** – žiadateľ dokazuje identitu určitou znalosťou (znalosť hesla),
b) **autentizácia predmetom** – žiadateľ dokazuje identitu predmetom (cestovný lístok, čipová karta),
c) **autentizácia biometrikou** – žiadateľ dokazuje identitu svojimi charakteristikami (odtlačok prstu, podpis).

Pre zvýšenie bezpečnosti autentizácie sa jednotlivé triedy kombinujú – multifaktorová autentizácia. Príkladom môže byť autentizácia do internet bankingu, kde uvádzame login a heslo a až po úspešnom overení sme dotazovaní na zadanie jednorázového hesla z SMS správy, poslanej na náš mobilný telefón.

1.1 Autentizácia znalosťou

Autentizácia znalosťou sa zakladá na poznatku, ktorý žiadateľ pozná. Tento poznatok majú žiadatelia uložený vo svojej pamäti. Najčastejšie má podobu reťazca numerických znakov – tzv. autentizačný kód („passcode“) alebo alfanumerických znakov – tzv. heslo („password“) a mal by byť ľahko zapamätateľný.

1.1.1 Login–Heslo

Pri autentizácii heslom je žiadateľ overený pomocou reťazca alfanumerických znakov, ktorý si sám volí alebo je mu pridelený. Heslo by malo byť aspoň 8 znakov dlhé s použitím veľkých a malých písmen, číslic a špeciálnych symbolov. Nemalo by obsahovať slovné výrazy, inak hrozí slovníkový útok. Podstatou slovníkového útoku je postupné skúšanie najčastejšie používaných hesiel a slovných spojení, pokiaľ nie je útočníkovi prístup povolený. Na strane kontroléru je tento útok možné obmedziť zablokovaním prístupu na určitý časový interval po zadaní niekoľkých nesprávnych hesiel.

Heslá nie je vhodné písať na papier, či ukladať ich do textových súborov. Každému aktívu by sme mali voliť rozdielne heslá, pretože pri náhodnom odhalení hesla bude môcť útočník pristupovať k viacerým aktívam. Vhodné je preto použiť tzv. heslovú frázu („passphrase“), ktorá pozostáva z dostatočne dlhého sledu slov. Heslovú frázu odvodíme z prvých písmen alebo znakov danej frázy, či vety, pričom dbáme na veľkosť písmen a na

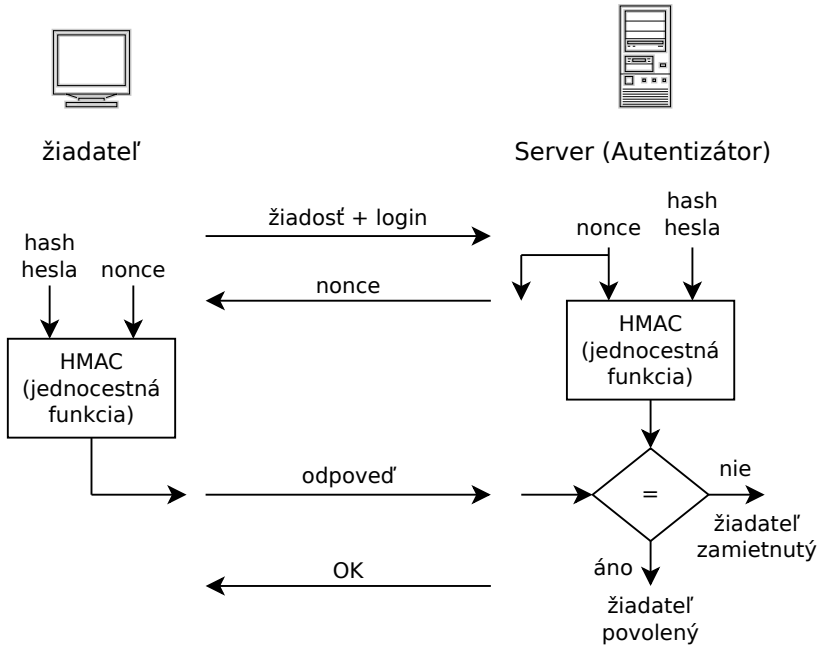
interpunkčné znamienka. Napr. heslová fráza z „Učím sa anglický jazyk už od 10 rokov!“ môže vyzerať nasledovne „Usajuo10r!“.

Na strane kontroléru sa heslá ukladajú s využitím jednocestnej hašovacej funkcie (napr. SHA-1), s ktorou sa pri autentizácii aj overujú. Problémom tohto uloženia je, že útočník si môže vytvoriť databázu hašov známych a slovníkových hesiel (prípadne všetkých možných kombinácií znakov) – tzv. rainbow tabuľka. Tomuto útoku je možné zabrániť mnohonásobnou iteráciou, kde sa haš vykoná viackrát, alebo tzv. zasolením, kde sa k heslu pridá určitý (variabilný) reťazec.

1.1.2 Výzva–Odpoveď

Autentizácia znalosťou typu Výzva–Odpoveď rozširuje autentizáciu typu Login–Heslo a považuje sa za bezpečnejšiu.

Princíp tejto autentizácie je nasledovný. Žiadateľ kontaktuje **autentizátor** (server) so žiadosťou, v ktorej uvedie svoje prihlasovacie meno (login). Autentizátor obratom pošle žiadateľovi náhodne vygenerovanú jednorázovú výzvu (tzv. **nonce**), kde nonce je číslom alebo reťazcom. Potom žiadateľ pošle odpoveď, ktorá pozostáva z hesla a verejného nonce, ktoré vstúpia do hašovacej funkcie alebo do HMAC (kde hash hesla je vstupný tajný kľúč). Autentizátor na základe uvedeného loginu vygeneruje kontrolný reťazec pomocou hashovacej funkcie z hashu hesla, uloženého v databáze, a poslaného nonce, ktorý porovná s odpoveďou žiadateľa. Ak sa kontrolný reťazec a odpoveď zhodujú, je žiadateľovi prístup povolený, v opačnom prípade zamietnutý. Pri útoku útočník odchytí žiadateľovu odpoveď a starý nonce, ktoré sú mu v budúcej autentizácii bezcenné. Tento proces je vyobrazený na obr. 1.1.



Obr. 1.1: Autentizácia znalosťou – výzva-odpoveď

1.1.3 Dôkaz nulovej znalosti

Autentizácia dôkazom nulovej znalosti sa zakladá na presvedčení autentizátora, že niečo pozná alebo niečo vie pomocou kryptografických metód bez toho, aby niečo prezradil poskytovateľovi (autentizátoru). Na tomto princípe sa zakladajú anonymné autentizácie, kde žiadateľ presvedčuje o svojej právoplatnosti na službu bez úniku svojej identity. V praxi sa tieto systémy zakladajú na asymetrickej kryptografii. Táto autentizácia je bezpečnejšia ako autentizácia typu Výzva-odpoveď.

1.2 Autentizácia predmetom

Pri autentizácii predmetom sa žiadateľ overuje pomocou fyzického predmetu (tzv. **tokenu**), ktorý žiadateľ vlastní. Predmetom je nejaký preukaz alebo pamäťové úložisko, ktoré má určité rysy a ochranné prvky, aby ich útočník nemohol falšovať.

Všeobecne rozlišujeme dva typy autentizácie predmetom:

a) predmety ako chránené úložisko s kryptografickou ochranou – šifrovanie dát na predmete (čipové karty, USB tokeny)
b) predmety s nechráneným úložiskom – vhodné k jednorázovej autentizácii (tzv. tickety, lístky), ktoré obsahujú prvky proti falzifikácii.

Výhodou autentizácie predmetom je zvýšená bezpečnosť, jednoduchá použiteľnosť, znižujúca sa cena. Nevýhodou je ich cena oproti softvérovému riešeniu autentizácie znalosťou.

1.2.1 Čipové karty

Čipové karty sa používajú v mobilných telefónoch (SIM karty), bankových systémoch (platobné karty) alebo umožňujú prístup do budov, či miestností. Používajú sa kontaktné karty s galvanickým rozhraním, bezkontaktné s rádiovým rozhraním, prípadne ich kombinácie – hybridy. Tieto karty (tzv. smart cards) obsahujú integrované obvody, ktorých súčasťou je procesor, pamäť, alebo kryptografické koprocesory (RSA, AES). V niektorých kartách bývajú implementované programovateľné jednotky, ktoré slúžia vývojárom na implementáciu nových kryptografických riešení – najčastejšie na platformách JAVA, či .NET.

1.2.2 USB tokeny

USB tokeny sú podobné čipovým kartám. Slúžia ako chránené úložisko súkromných kľúčov, certifikátov a iných tajných informácií. Účelom USB tokenov je útočníkom zabrániť prístupu a kopírovaniu chránených informácií, ku ktorým sa pristupuje najčastejšie cez PIN kód. Ich veľkosť býva okolo 32 kB, niekedy sa kombinujú s flash pamäťami (2 GB, 4 GB).

1.3 Autentizácia biometrikou

Autentizácia biometrikou sa zakladá na rozdielnosti charakteristík žiadateľov. Základnou požiadavkou je charakteristický rys žiadateľa, ktorý je jedinečný a nemenný, preto sa tento spôsob autentizácie používa výhradne u osôb. Výhodou je jej vysoká bezpečnosť využívaná v najprísnejších podmienkach. Nevýhodou biometrickej autentizácie je jej cena, pri lacnejších variantách menšia presnosť. Niektoré typy biometrických metód nemusia byť pre užívateľov príjemné.

Biometrickú autentizáciu rozdeľujeme do dvoch tried:

a) **fyziologické metódy** – fyziologické charakteristiky človeka (odtlačok prstu, tvár),
b) **behaviorálne metódy** – individuálnosť ľudského chovania (podpis).

1.3.1 Fyziologické metódy

Fyziologické metódy porovnávajú žiadateľov na základe ich odlišných fyziologických rysov, ktoré sú ich súčasťou. V súčastnej dobe sa používajú fyziologické biometriky založené na:

a) odtlačku prstov – individuálnosť tvaru papilárnych línií pomocou optických, kapacitných a ultrazvukových snímačov,
b) tváre – odfotografovanie tváre, vyhodnotenej na základe tvárovej metriky alebo charakteristickej tvári,
c) očnej dúhovke – individuálnosť rozmiestnenia a tvaru škvŕn na dúhovke
d) geometrii ruky – individuálne charakteristiky ruky (dĺžka a šírka prstov),
e) očnej sietnice – individuálne rysy cievneho riečiska očnej sietnice.

Zriedka sa vyskytujú biometrické autentizácie podľa DNA, podľa tvaru ucha, podľa ľudského pachu, podľa tvaru žlčného riečiska apod.

1.3.2 Behaviorálne metódy

Behaviorálne metódy rozlišujú žiadateľov pomocou ich rozdielnych ľudských chovaní.

Z behaviorálnych metód sa používajú biometriky založené na:

a) hlase – špecifická charakteristika rečníka (rýchlosť reči, frekvenčné spektrum hlasu),
b) spôsobu písaného podpisu – individuálnosť prevedenia ručného podpisu (dynamika, tlak na podložku),
c) spôsob písania na klávesnici – špecifické rysy zápisu určitej sekvencie znakov.

Kompletný rozpis autentizačných metód nájdete v [1]. Viac o kryptografií sa môžete dozvedieť v [2] a [3].

2 APLIKOVATELNOSŤ AUTENTIZÁCIE NA MOBILNÝ TELEFÓN

V dnešnej dobe nosíme mobilný telefón vždy pri sebe spolu s dokladmi. Používame ho nielen na telefonovanie a písanie textových správ, ale aj na prihlasovanie do elektronických bankových systémov, či prístupu do budov. Stáva sa čoraz výkonnejším a približuje sa bežným osobným počítačom.

Mobilné telefóny obsahujú základné prvky na ovládanie, ako sú tlačítka, klávesnice (častejšie už len virtuálne) a obrazovky. Taktiež majú rôzne senzory pre dodatočné ovládanie, akými sú:

- dotykové obrazovky,
- akcelerometre,
- gyroskopy,
- mikrofóny,
- kamery.

Pre ukladanie dát používajú rôzne typy úložísk, do ktorých sa okrem užívateľských dát môžu ukladať aj vlastné certifikáty, heslá a iné autentizačné informácie. Príkladom sú:

- SIM karty – pre uloženie kontaktov,
- vnútorné pamäte flash,
- SD karty – najčastejšie microSD.

Cellularná komunikácia, dátová a hlasová, prebieha na protokoloch uvedených v tab. 2.1. Pre blízku komunikáciu využívajú technológie IrDA (infračervený port), Bluetooth, WiFi. Najnovšie mobilné telefóny obsahujú technológiu NFC, ktorú používajú aj bezdrôtové smart-karty. Táto technológia slúži na prenášanie malých informácií (dát) pre iné zariadenia alebo ako autentizácia pre zabezpečené priestory a terminály.

Tab. 2.1: Prehľad GSM technológií

Technológia	**Rok vzniku**	**Prenosová rýchlosť**
GSM	1991	40 kbps
GPRS	2000	40 kbps (dáta)
EDGE	2003	400 kbps
HSDPA	2005	5,7 Mbps, 14,4 Mbps
HSUPA	2008	28/40 Mbps
LTE	2010	až 1 Gbps

Práve vlastnosti moderných mobilných telefónov (tzv. smartphone) nám umožňujú nahrádzať niektoré identifikačné predmety a údaje. Nahradzovaním preukazov a iných dôležitých informácii mobilnými telefónmi nastáva otázka bezpečnosti a ochrane týchto údajov. Dôležité je preto uzamykanie chránených častí mobilných telefónov pred neoprávnenými osobami a autentizácia užívateľov, ktorí do nich chcú vstupovať alebo ich používať. V opačnom

prípade by útočník z odcudzeného telefónu mohol tajné údaje použiť alebo by sa mohol vydávať za niekoho iného.

2.1 SIM karta

SIM (Subscriber Identity Module) karta je základným prvkom autentizácie v prostredí celulárnych sietí. Využitá je autentizácia pomocou predmetu, pričom je SIM karta dodatočne chránená PIN kódom.

Informácie, ktoré nesú SIM karty, sú použité na autentizáciu a identifikáciu účastníka mobilnej siete. Najdôležitejšie údaje sú:

a) **ICCID** – unikátne výrobné číslo karty (19 pozícií),
b) **IMSI** – unikátne identifikačné číslo cellulárnej siete: obsahuje číslo krajiny, číslo siete a číslo účastníka,
c) K_i – individuálny autentizačný kľúč (128-bitové číslo)
d) LAI – variabilné číslo oblasti.

SIM karty existujú vo viacerých vyhotoveniach, ktoré sú uvedené v tabuľke 2.2. Obsahujú mikrokontrolér, ktorého ROM pamäť je vo veľkosti 60–512 KB, RAM pamäť medzi 1 až 8 KB a EEPROM pamäť typicky okolo 16 až 512 KB. V ROM pamäti sa nachádza operačný systém, ktorý obsahuje sadu komunikačných prostriedkov pre komunikáciu s mobilným telefónom. Do EEPROM sa ukladajú certifikáty a užívateľské údaje (zoznam kontaktov, SMS). [4]

Tab. 2.2: Vyhotovenia SIM kariet

Vyhotovenie	**Dĺžka [mm]**	**Šírka [mm]**	**Hrúbka [mm]**
Plná veľkosť	85,60	53,98	0,76
mini-SIM	25,00	15,00	0,76
micro-SIM	15,00	12,00	0,76
nano-SIM	12,30	8,80	0,67

2.1.1 Autentizácia SIM karty

SIM karta býva zabezpečená PIN kódom. PIN kód je osobné identifikačné číslo s dĺžkou štyroch alebo ôsmich numerických znakov (v prípade SIM karty).

Proti uhádnutiu PIN kódu je jeho zadanie obmedzené na tri pokusy, po ktorých sa SIM karta zablokuje. V prípade uzamknutia SIM karty, nesprávnym zadaním PIN kódu, je nutné pre odblokovanie zadať PUK kód (osobný odblokovací kód), ktorý zvyčajne býva dlhší ako PIN kód. Na PUK kód sa tiež vzťahuje obmedzenie zadania, a to na 10 pokusov, po ktorých sa SIM karta trvalo zablokuje. Zablokovanie SIM karty alebo nesprávne zadanie PIN kódu môže v niektorých prípadoch viesť aj k zablokovaniu mobilného telefónu.

PIN kód sa vyžaduje vždy pri zapnutí mobilného telefónu alebo po prebudení z letového režimu (ak telefón túto funkciu podporuje), teda v prípade, ak chceme používať

telekomunikačné služby a pristupovať do údajov SIM karty. Pravdepodobnosť náhodného uhádnutia PIN kódu je vyobrazená v tabuľke 2.3. PIN kód sa môže použiť aj na blokovanie prístupu do telefónu, pri ktorom je dĺžka tohto kódu variabilná v závislosti od užívateľa. Menej často sa používa PIN kód väčší ako 8 znakov, kde je zapamätanie kódu náročné a ľahko zabudnuteľné.

Tab. 2.3: Pravdepodobnosť uhádnutia PIN kódu

Dĺžka kódu	3	4	6	8
Pravdepodobnosť [%]	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-8}$

2.1.2 Autentizácia v GSM sieti

Mobilné telefóny sa pri pripojení do celulárnej siete musia autentizovať, aby mohli využívať telekomunikačné služby. Pri autentizácii telefón vyšle požiadavku na prístup do siete spolu s IMSI údajom. Požiadavka je smerovaná pomocou mobilných centier prepínačov (MSC) do domáceho registra operátora (HLR). V tomto registri sa nachádzajú všetky informácie účastníkov zahŕňajúce IMSI, aktuálnu pozíciu alebo aktivované služby. Autentizácia je vyžadovaná pri zmenách polohy (pripojenie mobilného zariadenia do inej bunky) alebo na základe určitých udalostí.

V prípade, že sa odoslaný IMSI údaj nachádza v databáze HLR, prepošle sa požiadavka do Autentizačného centra (AuC). Autentizačné centrum nájde kľúč K_i na základe preposlaného IMSI. Kľúč K_i je 128-bitové číslo, ktoré sa nachádza len na SIM karte a v Autentizačnom centre operátora. AuC potom vygeneruje 128-bitové číslo nazývané RAND. RAND a K_i vstúpia do A3 enkrypčného algoritmu, ktorého výstupom je 32-bitová podpísaná odozva (SRES). Ďalej sa vygeneruje 64-bitový šifrovací kľúč K_c pomocou algoritmu A8, ktorého vstupom je RAND a K_i. K_c kľúč sa využije pri šifrovaní a dešifrovaní pomocou A5 enkrypčného algoritmu.

Vygenerované čísla RAND, SRES, a K_c sa z AuC prepošlú do HLR a následne do MSC. MSC si najprv uloží K_c a SRES, ale mobilnej stanici pošle RAND číslo na základe požiadavky. Mobilný telefón vygeneruje, podobne ako AuC, kľúče SRES a K_c pomocou algoritmov A3 a A8, ktorých vstupom je dvojica K_i (uložené na SIM karte) a prijatý RAND. Kľúč K_c si mobilný telefón uloží do SIM karty a SRES pošle do siete. MSC následne porovná údaj SRES prijatý z autentizačného centra a ten prijatý z mobilného telefónu. V prípade zhody je mobilné zariadenie úspešne autentizované.

Keď bolo zariadenie autentizované, MSC pošle uložený kľúč K_c do základňovej stanice, na ktorú je mobilné zariadenie pripojené. Mobilný telefón dostane požiadavku na prepnutie sa do šifrovaného režimu. Všetky ďalšie dáta medzi mobilným zariadením a základňovou stanicou sú posielané zašifrované pomocou A5 algoritmu s použitím kľúča K_c. Šifrovací kľúč K_c nie je prenášaný v bezdrôtovej sieti účastníkov, je uložený len v základňovej stanici a na SIM karte účastníka.

Viac o autentizácii a bezpečnosti v mobilných sieťach GSM, GPRS a UMTS sa dočítate v [5].

2.2 Autentizácia heslom

Mobilné telefóny obsahujú osobné dáta, ako sú telefónne kontakty, elektronické peňaženky apod. Ak chceme tieto údaje ochrániť, môžeme využiť dodatočnú autentizáciu heslom. V nej platia rovnaké zásady voľby hesla popísané v sekcii 1.1.1.

Heslo obsahuje numerické alebo alfanumerické znaky. Vyžaduje sa buď pri prístupe do chránených dát, aplikácie s dátami, alebo sa ním zablokuje celý telefón. V prípade zablokovania celého telefónu heslom je možné nastaviť trvácnosť sedenia, kde po vypršaní sedenia je heslo nutné znovu zadať. Taktiež je možné dáta ochrániť po viacerých neúspešných pokusoch zadania hesla ich vymazaním.

2.2.1 Bezpečnosť hesla

Heslo je považované za bezpečné, ak je minimálne 8 znakov dlhé a obsahuje rôzne špeciálne znaky, veľké a malé písmená a čísla. Môžeme ho vytvoriť pomocou generátora hesiel alebo môžeme vytvoriť heslovú frázu (viď sekcia 1.1.1).

Tab. 2.4: Pravdepodobnosť uhádnutia hesla

	Pravdepodobnosť uhádnutia [%]		
Použité znaky	**4-znakové**	**8-znakové**	**10-znakové**
a–z (26)	$2{,}19 \cdot 10^{-6}$	$4{,}79 \cdot 10^{-12}$	$7{,}08 \cdot 10^{-15}$
a–z, A–Z (52)	$1{,}37 \cdot 10^{-7}$	$1{,}87 \cdot 10^{-14}$	$6{,}92 \cdot 10^{-18}$
a–z, A–Z, 0–9 (62)	$6{,}77 \cdot 10^{-8}$	$4{,}58 \cdot 10^{-15}$	$1{,}19 \cdot 10^{-18}$
vytlačiteľné ASCII znaky (95)	$1{,}23 \cdot 10^{-8}$	$1{,}51 \cdot 10^{-16}$	$1{,}67 \cdot 10^{-20}$

Pravdepodobnosť náhodného uhádnutia hesla môžeme vyjadriť pomocou

$$P_{\text{hesla}} = \frac{1}{z^m}, \tag{2.1}$$

kde z je počet možných znakov a m je počet miest (dĺžka hesla) – z čoho je vidieť, že možnosť uhádnutia klesá exponenciálne so zvyšujúcou sa dĺžkou hesla. Vyčíslené príklady pravdepodobnosti uhádnutia hesla sú uvedené v tabuľke 2.4.

2.2.2 Možnosti uloženia hesla na platforme Android

Platforma Android neposkytuje mnoho bezpečných spôsobov ako uložiť cenné informácie a heslá. Avšak môžeme využiť šifrovacie nástroje a algoritmy a uložiť tieto informácie do nezabezpečených oblastí systému.

Vybrať môžeme z nasledovných spôsobov:

a) uloženie hash-u hesla,
b) použitie šifrovacích algoritmov (AES),
c) uloženie na zabezpečený súborový systém,
d) uloženie pomocou manažéra online účtov,
e) uloženie do credential storage – úložisko privátnych kľúčov a certifikátov.

Pokiaľ chceme využiť možnosť uloženia hesla na zabezpečený súborový systém, musíme mať android vybavený ovládačom pre tento typ systému, čo vyžaduje aj tzv. root telefónu s OS Android. Taktiež je nutné vlastniť SD kartu a vytvoriť na nej partíciu zabezpečeného súborového systému, prípadne ju vytvoriť na dostupnej pamäti zariadenia.

Uloženie do credential storage

Andorid od verzie 4.0 (ICS) podporuje prístup do credential storage jednotlivým aplikáciám. V nižších verziách bol tento typ úložného systému prístupný len pre systémové nástroje na ukladanie WiFi a VPN hesiel a certifikátov.

Credential storage systému Android je implementovaný ako natívna služba linuxu, ktorú je možné obsluhovať pomocou dostupného API. Kľúče v tomto úložisku sú šifrované 128-bitovým AES master kľúčom. Master kľúč je derivovaný z hesla alebo PIN kódu zadaného užívateľom v hlavných nastaveniach systému. Uložené kľúče sú ukladané ako jednotlivé súbory, kde každá aplikácia má prístupné a viditeľné len svoje položky. V prípade root-nutého telefónu sú súbory z credential storage pri odcudzení bezcenné, bez master kľúča sa veľmi ťažko dekódujú.

Pristúpiť do credential storage môže aplikácia až po odblokovaní prístupu. Pre odblokovanie sa vytvorí *Intent* `com.android.credentials.UNLOCK`, ktorý je súčasťou API. *Intent* vytvorí dialógové okno, kde užívateľ zadá master kľúč – ide mimo vlastnej aplikácie. Po zadaní správneho hesla je credential storage pre aplikáciu prístupné a je možné z neho čítať a doňho zapisovať.

Kompletný popis funkčnosti Credential Storage nájdete v [6].

2.3 Autentizácia gestom

Autentizácia gestom sa považuje za bezpečnejšiu ako je autentizácia heslom. Užívateľ si zvolí gesto na displeji mobilného telefónu, ktoré bude musieť zadávať pri odblokovaní obrazovky alebo prístupe do chránenej aplikácie. Nutnosťou je vlastniť telefón s dotykovou obrazovkou – kapacitnou alebo odporovou.

Gesto je možné zadať dvoma spôsobmi:
a) spájaním bodov priamkami,
b) nakreslením krivky.

2.3.1 Gesto spojením bodov

Tento spôsob autentizácie je typický pre platformu Android. Gestom sa spájajú body jedným ťahom, rozmiestnených na matici o veľkosti 3×3. Zohľadňuje sa aj smer spojenia týchto bodov, pričom je možné spojiť body nachádzajúce sa na krajoch matice – priamka však nesmie viesť cez žiadny bod. Najmenej je možné spojiť 4 body a najviac 9 bodov (max. počet bodov). Výhodou tohto gesta je jednoduchosť, pomocné navádzanie a vyššia bezpečnosť oproti heslovej autentizácii.

Tab. 2.5: Počet možností zadania gesta spojením bodov

Počet spojených bodov	**Možnosti**
4	1 624
5	7 152
6	26 016
7	72 912
8	140 704
9	140 704
Celkovo:	389 112

2.3.2 Gesto nakreslením krivky

Ďalším spôsobom zadania gesta je pomocou nakreslenia krivky. Krivka sa kreslí na dotykovú obrazovku bez navádzania fixnými bodmi. Nakreslená krivka sa navzorkuje a porovná sa s krivkami uloženými v telefóne. Vyhodnocujú sa inštancie dvoch vektorov na základe euklidovskej metriky alebo kosínusovej vzdialenosti. Vypočítaná vzdialenosť musí byť väčšia ako zvolený prah, aby mohol byť užívateľ autentizovaný. V prípade viacerých uložených gest sa vyberie gesto s najväčšou podobnosťou.

2.4 Autentizácia hlasom

Autentizácia hlasom je založená na overení neznámeho rečníka, ktorý sa vydáva za osobu známu. Overuje sa podobnosť hlasu neznámej osoby s registrovanými rečníkmi na základe príznakov reči. Patrí do skupiny biometrických metód (1.3 Autentizácia biometrikou).

2.4.1 Verifikácia a identifikácia rečníka

Verifikácia rečníka

Verifikácia rečníka (osoby), ktorý chce pristúpiť do zabezpečenej (uzamknutej) oblasti, prebieha deklarovaním totožnosti a vyslovením príhovoru. Vyslovená reč sa spracuje a reprezentuje sa pomocou príznakov.

Reprezentácia hlasu sa porovná s reprezentáciou uloženou v databáze (vyberie sa na základe deklarovanej totožnosti) a určí sa tzv. **verifikačná miera**, ktorá udáva mieru

podobnosti alebo mieru vzdialenosti medzi oboma reprezentáciami hlasu. Ak je získaná miera väčšia ako verifikačný prah (dopredu zvolený), je žiadateľ úspešne overený.

Identifikácia rečníka

Pri identifikácii rečníka je úlohou zistenie totožnosti zo skupiny známych registrovaných rečníkov. Budeme predpokladať, že neznámy rečník patrí do skupiny referenčných rečníkov – tzv. **identifikácia v uzavretej množine**. Potom neznámemu rečníkovi priradíme identitu jedného z registrovaných rečníkov, ktorej reprezentácia hlasu je čo najpodobnejšia.

Ak budeme predpokladať, že neznámy rečník nemusí patriť do skupiny známych rečníkov, jedná sa o tzv. **identifikáciu v otvorenej množine**. Identifikácia potom prebieha v dvoch fázach. V prvej sa bude predpokladať, že neznáma osoba bude patriť medzi známych rečníkov – prevedie sa identifikácia v uzavretej množine. V druhej fáze je získaná totožnosť, tá ktorej hlas neznámeho rečníka je najpodobnejší hlasu známej osoby, pokladaná za deklarovanú identitu a pristúpi sa k verifikácii (viď Overenie rečníka). Ide o kombináciu identifikácie v uzavretej množine a verifikácie rečníka.

Pracovné režimy

Systém rozpoznávania rečníka pracuje v niekoľkých režimoch:

a) **rozpoznávania** – v ktorom sa vykonáva samotná verifikácia a identifikácia rečníka,
b) **trénovania** – kde sa berú vzorky hlasu známeho rečníka a pridávajú sa do databázy registrovaných rečníkov (pri identifikácii na otvorenej množine sa vytvára aj databáza podvodníkov),
c) **testovania** – kde sú testovacie vzorky známeho rečníka získavané pre stanovenie hodnoty verifikačného prahu, najlepšie vo všetkých typoch prostredí,
d) **vyhodnocovania** (evaluácie) – využívaný k ohodnoteniu systému rozpoznávania, bez zmeny parametrov.

2.4.2 Typy systémov

Systémy rozpoznávania môžeme rozdeliť do skupín podľa rečového materiálu dostupného pri trénovaní a pri rozpoznávaní. Prvým typom je **textovo závislé rozpoznávanie**, pri ktorom rečník pri identifikácii vyslovuje rovnaký príhovor ako príhovor, ktorý uviedol (nahral) pri trénovaní systému. Príhovor môže byť pokladaný za heslo, ktoré má každý užívateľ rôzne – osobné heslo, alebo je heslo rovnaké pre všetkých známych rečníkov – všeobecné heslo. Taktiež sa používa **rozpoznávanie s pevným slovníkom**, kde príhovor pri rozpoznávaní má obsahovať slová z testovacieho príhovoru. Rečník môže vysloviť súbor číslic vyobrazených v náhodnom poradí, ktoré uviedol pri trénovaní systému. Pokiaľ rečník nie je ochotný vysloviť daný príhovor, je využiteľný typ **rozpoznania závislý na udalosti**. Z vysloveného prejavu sa vyberú a analyzujú fonetické javy. Príkladom je rozpoznanie na základe samohlások.

Ďalej môžeme systémy rozpoznania rečníka zadeliť do skupín podľa výzvy. Jedná sa o **systémy s textovou výzvou**, v ktorej neznáma osoba vyslovuje príhovor zobrazený na obrazovke zariadenia. Výzvou je jednoduchá veta, ktorú rečník vysloví, alebo pokyn vyslovenia[1]. **Systémy s hlasovou výzvou** pracujú na obdobnom spôsobe ako systémy s textovou výzvou, kde sa výzva prehráva na reproduktoroch. Výhodou tohto systému je, že rečník sa snaží napodobňovať zachytenú zvukovú výzvu, čo systému umožňuje ľahšie a presnejšie rozpoznanie. Ak nie je od rečníka očakávaná žiadna konkrétna výzva, ide o **systémy bez výzvy**.

2.4.3 Ohodnotenie činnosti systému

Systém rozpoznania rečníka vo vyhodnocovacom režime pracuje rovnako ako v režime rozpoznávania, s tým rozdielom, že máme dostupné informácie o správnych výsledkoch. Vyhodnocovací režim je rozdielny pri systémoch verifikácie, systémoch identifikácie v uzavretej a v otvorenej množine. Vyhodnocuje sa pri určitých podmienkach (napr. množstvo použitých dát, použitý mikrofón), ktoré je treba uviesť.

Ohodnotenie činnosti systému verifikácie

Pri systéme verifikácie, kde sa vyhodnocuje miera podobnosti medzi reprezentáciou hlasu žiadateľa a hlasu registrovaného rečníka, rozlišujeme štyri možné situácie. Ak sa neznámy rečník vydáva za seba a miera podobnosti bude väčšia ako verifikačný prah, ide o **správne prijatie**. Ak sa osoba vydáva za seba, ale miera podobnosti bude menšia ako verifikačný prah, ide o **nesprávne odmietnutie**. V prípade, že sa neznáma osoba vydáva za niekoho iného, než v skutočnosti je, a miera podobnosti bude väčšia ako verifikačný prach, jedná sa o **nesprávne prijatie**. Ak sa však neznámy rečník vydáva za niekoho iného a miera podobnosti bude po vyhodnotení menšia ako verifikačný prah, budeme hovoriť o **správnom odmietnutí**.

Pri hodnotení činnosti systému sa využíva miera súvisiaca s počtom nesprávnych prijatí a nesprávnych odmietnutí. Hodnotí sa **pomerný počet chýb nesprávneho prijatia** $R_{\mathrm{FA}}(\Theta)$ a **pomerný počet chýb nesprávneho odmietnutia** $R_{\mathrm{FR}}(\Theta)$, kde Θ je daný verifikačný prah.

Pomerný počet chýb nesprávneho prijatia je odhadom pravdepodobnosti, že systém bude akceptovať podvodníka. Počíta sa podľa vzťahu

$$R_{\mathrm{FA}}(\Theta) = \frac{n_{\mathrm{FA}}(\Theta)}{n_{\mathrm{podv}}}, \tag{2.2}$$

kde $n_{\mathrm{FA}}(\Theta)$ je počet pokusov, v ktorých systém pri verifikačnom prahu Θ vyhodnotil podvodníka ako osobu, za ktorú sa podvodník vydával, a n_{podv} je počet verifikačných pokusov, v ktorých bol žiadateľ podvodníkom.

[1] Pokyn vyslovenia môže byť napr. v tvare „Vyslovte štyri ľubovolné číslice“.

Pomerný počet chýb nesprávneho odmietnutia je odhadom pravdepodobnosti, že systém odmietne správneho rečníka. Je vyjadrený pomocou vzťahu

$$R_{\mathrm{FR}}(\Theta) = \frac{n_{\mathrm{FR}}(\Theta)}{n_{\mathrm{sp\_ref}}}, \tag{2.3}$$

kde $n_{\mathrm{sp\_ref}}$ je počet pokusov, v ktorých bol žiadateľom správny referenčný rečník, a $n_{\mathrm{FR}}(\Theta)$ udáva počet pokusov, v ktorých systém pri verifikačnom prahu Θ správneho rečníka odmietol.

Činnosť systému verifikácie môžeme ohodnotiť jedným číslo, tzv. **mierou rovnosti chýb** R_{ERR}, ktorú definujeme pomocou vzťahu

$$R_{\mathrm{ERR}} = R_{\mathrm{FR}}(\Theta) = R_{\mathrm{FA}}(\Theta). \tag{2.4}$$

Zo vzťahu je zjavné, že musíme zvoliť taký verifikačný prah Θ, pri ktorom sa bude pomerný počet chýb nesprávneho prijatia a nesprávneho odmietnutia rovnaký.

Ohodnotenie činnosti systému identifikácie v uzavretej množine

V systéme pracujúcom na identifikácii v uzavretej množine môžu nastať len dve situácie. **Správna identifikácia** nastane, ak systém identifikuje neznámeho rečníka správne. **Nesprávna identifikácia** nastane, ak systém identifikuje neznámeho rečníka ako niekoho iného, ktorým v skutočnosti nie je.

Udáva sa **miera úspešnosti identifikácie** R_{S} definovaná vzťahom

$$R_{\mathrm{S}} = \frac{n_{\mathrm{spr}}}{n_{\mathrm{celk}}}, \tag{2.5}$$

kde n_{celk} je celkový počet identifikačných pokusov a n_{spr} je počet pokusov, v ktorých bol neznámy rečník správne identifikovaný. Alebo sa uvádza **miera neúspešnosti identifikácie** R_{E}, ktorú môžeme zapísať ako

$$R_{\mathrm{E}} = \frac{n_{\mathrm{chyb}}}{n_{\mathrm{celk}}}, \tag{2.6}$$

kde n_{chyb} udáva počet pokusov, v ktorých došlo k chybnej identifikácii. Mieru úspešnosti identifikácie je možné previesť na mieru neúspešnosti a naopak, a to

$$R_{\mathrm{E}} = 1 - R_{\mathrm{S}}. \tag{2.7}$$

Ohodnotenie činnosti systému identifikácie v otvorenej množine

V prípade systému pracujúceho na identifikácii v otvorenej množine ide o kombináciu systému identifikácie na uzavretej množine a systému verifikácie rečníka, teda môžu nastať všetky situácie obsiahnuté v týchto systémoch. Možné výsledky sú vyobrazené v tab. 2.6.

Tab. 2.6: Možné výsledky systému identifikácie v otvorenej množine v prípade, že referenčnými rečníkmi sú Eva a Pavol

<table>
<tr><th>Neznámy rečník</th><th>Rozhodnutie systému</th><th colspan="2">Výsledná situácia</th></tr>
<tr><td rowspan="2">Adam</td><td>Neznámy rečník nepatrí do referenčných rečníkov</td><td colspan="2">správne odmietnutie</td></tr>
<tr><td>Neznámy rečník je Peter</td><td colspan="2">nesprávne prijatie</td></tr>
<tr><td rowspan="3">Eva</td><td>Neznámy rečník nepatrí do referenčných rečníkov</td><td colspan="2">nesprávne odmietnutie</td></tr>
<tr><td>Neznámy rečník je Eva</td><td rowspan="2">správne prijatie</td><td>správna identifikácia</td></tr>
<tr><td>Neznámy rečník je Pavol</td><td>nesprávna identifikácia</td></tr>
</table>

2.4.4 Spoľahlivosť rozpoznania rečníka

Systém rozpoznania rečníka patrí do skupiny biometrických metód, kde sa hlas človeka rozpoznáva na základe behaviorálnych biometrík. Výsledok vyhodnotenia závisí od prostredia, v ktorom sa osoba nachádza, od nálady rečníka a pod. Aby bola dosiahnutá určitá úroveň spoľahlivosti, je nutné dodržať nasledujúce podmienky:

a) rečník nesmie meniť svoj hlas,
b) podmienky pri nahrávaní hlasu a techniky spracovania sú známe,
c) nahrávaný hlas sa nachádza v rovnakých podmienkach, ako boli podmienky pri trénovaní systému,
d) verifikačný prah stanovený samostatne pre každú aplikáciu – na základe testovacích dát.

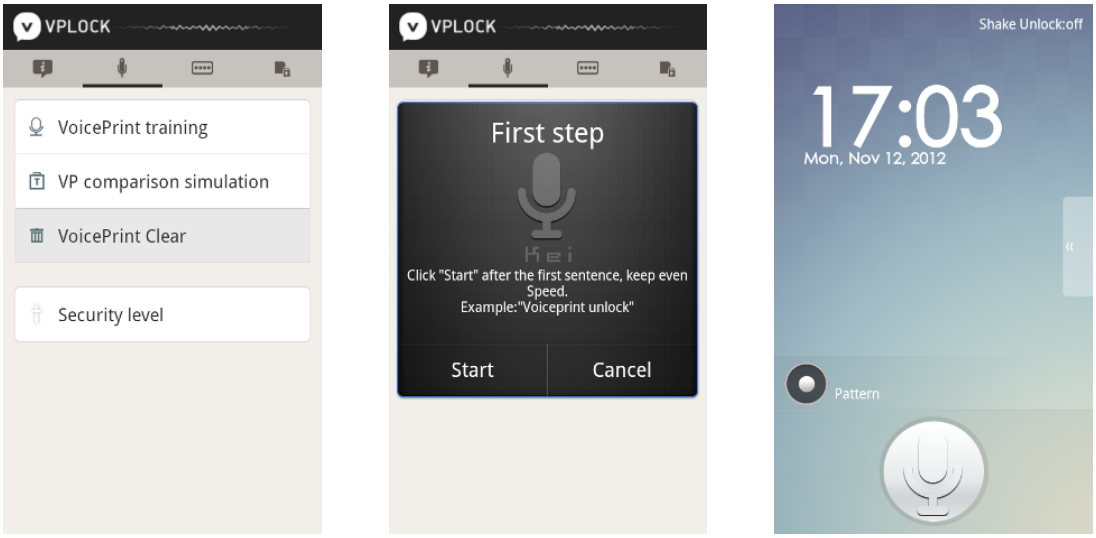
Súčasné metódy pri textovo závislých systémoch verifikácie rečníka dosahujú mieru rovnosti chýb až 0,5 % v prípade kvalitných nahrávok a u textovo nezávislých dosahujú mieru rovnosti chýb približne 2 %. Ak však použijeme lacnejšie nahrávacie zariadenia, akými sú aj mobilné telefóny, dosiahneme pri textovo závislých systémoch 2 % a pri textovo nezávislých až 10 % miery rovnosti chýb. Systémy identifikácie rečníka dosahujú úspešnosť v rozmedzí 60–99 %, kde najväčšie úspešnosti sa dosahujú pri veľmi kvalitných nahrávkach.

2.4.5 Dostupná aplikácia (VPLock)

Platforma Android natívne neposkytuje uzamykanie a autentizáciu na základe hlasu, avšak v *Google Play* existuje zopár aplikačných riešení. Jedným z riešení autentizácie hlasom je aplikácia *VPLock Free*, ktorá uzamyká prístup do mobilného telefónu a na jeho odblokovanie používa viacero typov autentizácií.

Aplikácia *VPLock* využíva textovo závislý systém verifikácie užívateľa. Všetky potrebné úkony sa v nej vykonávajú v nastaveniach (Obr. 2.1a), ktoré sú chránené heslom. Prvým krokom pri autentizácii hlasom je vytvorenie hlasovej reprezentácie rečníka, čo sa vykoná pri trénovaní systému („VoicePrint training“, Obr. 2.1b) v štyroch krokoch. Druhým krokom je overenie správnosti hlasovej reprezentácie v simulácii zhody hlasových reprezentácií

(„VP comparison simulation“). Ďalej je možné uviesť striktnosť rozhodovania zhody – približná voľba verifikačného prahu. Po nastavení aplikácií je nutné spustiť príslušnú službu, ktorá bude obsluhovať odomykanie telefónu pri uzamknutej obrazovke (Obr. 2.1c).



(a) (b) (c)

Obr. 2.1: Aplikácia VPLock: a) menu nastavenia a trénovania hlasových vzoriek, b) prvý krok trénovania systému, c) uzamknutá obrazovka

2.4.6 Zhodnotenie

Systémy autentizácie na základe hlasu rečníka je možné využívať na výkonnejších prenosných zariadeniach (mobilných telefónoch) s využitím dostupného mikrofónu. V závislosti na dostupnom hardvéru a výkone zariadenia je možné zvoliť určitú technológiu rozpoznávania. Využije sa systém identifikácie na otvorenej množine, kde bude referenčným rečníkom len jedna osoba[2].

Tieto systémy pri prenosných zariadeniach dosahujú malú úspešnosť, a to najmä z dôvodu šumu a rozdielnosti prostredí, v ktorých sa užívatelia nachádzajú. Problém správneho vyhodnotenia a autentizácie môže nastať napr. v prípade, ak má daná osoba zastretý alebo zachrípnutý hlas.

Viac o autentizácii rozpoznávania rečníka nájdete v [7].

2.5 Autentizácia tvárou

Autentizácia tvárou patrí do skupiny **autentizácie biometrikou** (viď 1.3.2). Používa sa aj na identifikáciu občanov, napr. pri pasových kontrolách alebo letiskách. Oproti ostatným biometrickým metódam je metóda autentizácie tvárou prirodzená, jednoduchá, a nie

[2]V prípade, že systém podporuje viacerých užívateľov (Android 4.2), je možné mať viac referenčných rečníkov.

je užívateľom nepríjemná v porovnaní s metódou autentizácie pomocou očnej dúhovky. Pri verifikácii sa porovnáva aktuálne spravená snímka so snímkou uloženou v zariadení, kde sa snímky môžu porovnávať v čase („realtime“) alebo ako jednorázová snímka.

2.5.1 Rozpoznávanie tváre

Pri rozpoznávaní tváre je dôležitá kvalita snímky, kde tvár v trojrozmernej dimenzii, podliehajúca expozícii, tieňom a výrazu, transformujeme do dvojrozmerného obrázku.

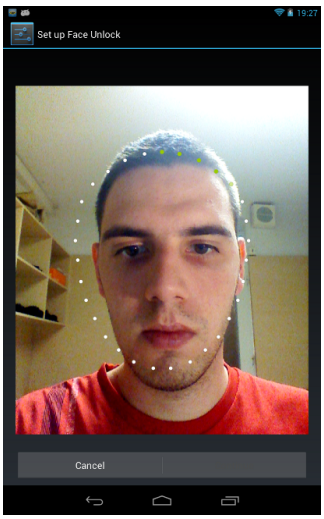
Systém rozpoznania tváre pozostáva zo štyroch častí:

a) detekcia a sledovanie tváre,
b) zarovnanie snímky tváre,
c) získanie znakov tváre,
d) porovnávanie.

Pomocou detekcie tváre sa zistí pozícia tváre na snímke, na základe ktorej sa odstráni pozadie snímky. Následne sa tvár zarovná, normalizuje, upraví na určitú požadovanú veľkosť, prípadne sa natočí to zvislej polohy. Taktiež sa snímka riadne nasvieti a prevedie do stupňov šedej. Potom sa vyberú určité znaky tváre, akými sú oči, ústa, nos, obočie a krivky tváre. Tieto znaky sa porovnajú so znakmi uloženými v zariadení, na základe ktorých sa vyhodnotí zhoda a užívateľovi je prístup povolený alebo zmietnutý.

Rozpoznávanie a autentizáciu pomocou tváre rozpisuje [8].

2.5.2 Odomykanie telefónu pomocou tváre



Obr. 2.2: Vytváranie snímku tváre pre odblokovanie tvárou

Platforma Android od verzie 4.0 obsahuje uzamykanie telefónu a jeho odblokovanie pomocou tváre užívateľa. Hlavnou podmienkou tohto typu autentizácie je vlastníctvo mobilného zariadenia s prednou kamerou. Odomykanie pomocou tváre je ale menej bezpečnejšie ako pomocou gesta, PIN kódu, či hesla. Ak sa nájde osoba s rovnakými črtami tváre,

je veľmi pravdepodobné, že sa telefón jednoducho odblokuje.

Znaky tváre sa nahrávajú v globálnych nastaveniach systému android pod položkou *Zabezpečenie*. Tvár je najlepšie nasnímať vo vnútorných podmienkach s prirodzeným osvetlením, pričom telefón je nutné držať približne na úrovni očí. Samotné odblokovanie prebieha v reálnom čase, takže stačí kameru nasmerovať tak, aby sa tvár vošla do vyhradenej oblasti.

2.6 Zhodnotenie autentizačných metód

Prehľad a zhodnotenie autentizačných metód aplikovateľných na mobilné telefóny je uverejnený v tabuľke 2.7. Najlepšou a najbezpečnejšou metódou autentizácie, použiteľnou na mobilných telefónoch, je autentizácia pomocou nakreslenia krivky. Vyžaduje malé nároky na vybavenie a hardvér telefónu a je približne na 100 % úspešná.

Tab. 2.7: Subjektívne zhodnotenie autentizačných metód aplikovateľných na mobilné telefóny

Metóda	**Bezpečnosť**	**Úspešnosť**	**Náročnosť**	**Potrebné vybavenie**
PIN kód	malá	—	nízka	vstupné periférium
Heslo	nízka/malá	—	nízka	vstupné periférium
Gesto spojením bodov	stredná	$\sim 100\,\%$	nízka	vstupné periférium
Krivkové gesto	vysoká	90–100 %	stredná	dotykové periférium
Hlas	stredná	60–80 %	vysoká	mikrofón
Tvár	malá	60–90 %	vysoká	kamera

3 MOŽNOSTI AUTENTIZAČNÝCH SCHÉM NA PLATFORME ANDROID

Android je otvorený operačný systém určený najmä pre mobilné zariadenia, ktorého jadrom je Linuxový kernel. Podporuje niekoľko procesorových architektúr, akými sú ARM, MIPS, či x86. Užívateľské aplikácie sú spúšťané pod jednotlivými systémovými užívateľmi vo virtuálnych prostrediach – tzv. Dalvik Virtual Machine. Aplikácie sú napísané v jazyku Java a využívajú aplikačný framework Androidu (AAF), ktorý obsahuje aj „orezané“ triedy z Javy. Balíček aplikácie má príponu APK (v Jave JAR), čo je komprimovaný archív kompilovaných tried, kódu a rôznych zdrojov (obrázky, texty).

3.1 Stavba a komponenty aplikácií

Aplikácie systému Android vytvárajú tzv. **aktivity**, ktoré obsluhujú užívateľské interakcie alebo vykonávajú určité operácie. Vstupným bodom aplikácie je hlavná aktivita – čo je trieda, ktorá dedí `Activity` triedu frameworku, definovaná je v konfiguračnom súbore aplikácie, v tzv. Manifest.

Správu aktivít režíruje operačný systém, ktorý jednotlivé aktivity, prípadne celé aplikácie, vypína alebo zahadzuje, v závislosti od dostupných systémových zdrojov. Trieda aktivity obsahuje metódy, ktoré môžu na tieto udalosti, v závislosti na stave, vhodne zareagovať, napr. uložením potrebných dát pri jej stopnutí či zahodení. Ak chce aktivita vytvoriť alebo prejsť do inej aktivity (napr. pri zobrazení nového okna) vyvolá `Intent`, čo je hlavný prostriedok komunikácie medzi dvomi aktivitami.

Ďalším komponentom aplikácie sú **služby** (trieda `Service`) určené na vykonávanie operácií na pozadí, napr. prehrávanie hudby alebo sťahovanie súboru. Služby dovoľujú aplikáciám zdieľanie funkcií cez dlhodobé spojenia (FTP, HTTP). Na rozdiel od aktivít sú služby zahodené zvyčajne len pri nedostatku pamäti.

Poslednými komponentami aplikácie sú tzv. **Content provider**-y a **Broadcast receiver**-y. Content provider-y poskytujú dáta na vyžiadanie, pracujú na rovnakom princípe ako RESTful webové služby. Broadcast receiver-y sú komponenty, ktoré načúvajú na určité udalosti systému alebo prostredia, napr. ak klesne hladina batérie pod nízku úroveň.

Užívateľské rozhranie je možné definovať pomocou vytvorenia jednotlivých prvkov priamo v kóde aktivity pri jej vytvorení alebo je možné tieto prvky vytvoriť v XML súbore, ktorý je súčasťou zdrojov aplikácie. Komponenty užívateľského rozhrania (tlačítka, textové polia, atď.) sú nazývané ako `View` objekty a sú zaraďované do `ViewGroup` skupín. Aktivita k nim môže pristupovať pomocou ich ID identifikátoru.

Viac o stavbe aplikácií sa môžete dočítať v [9] alebo na adrese <`http://developer.android.com/guide/components/index.html`>.

3.2 Možnosti implementácie

3.2.1 Operácie nad veľkými číslami a modulárna aritmetika

Autentizačné a kryptografické metódy využívajú modulárnu aritmetiku nad veľkými číslami, napr. pri generácii veľkých náhodných prvočísel. Veľké čísla nie sú natívne podporované operačným systémom ale frameworkom Androidu (Javy) a sú dostupné v štandardnej knižnici Javy `java.math`. Pre celé čísla sa využíva trieda `BigInteger` a pre desatinné `BigDecimal`.

Nad číslami typu `BigInteger` a `BigDecimal` je možné robiť nasledovné matematické operácie:

- sčítanie (`add`),
- negácia (`negate`),
- násobenie (`multiply`),
- delenie (`divide`, `divideAndRemainder`),
- modulo a inverzné modulo (`mod`, `modInverse`),
- umocnenie (`pow`),
- umocnenie s modulom (`modPow`),
- logické operácie (`and`, `or`, `xor`, `not`).

Operáciu umocnenia s modulom (`modPow`), vyjadrenú v Jave ako `c.modPow(a,b)`, môžeme matematicky zapísať vzorcom

$$c^a \mod b, \tag{3.1}$$

kde čísla a, b, c sú typu `BigInteger`.

3.2.2 Symetrické šifrovanie

Kryptografické metódy pre platformu Android je možné nájsť v knižniciach `java.security` a `javax.crypto`. Pre symetrické šifry je možné použiť abstraktný model `Cipher`, ktorý v platforme Android obsahuje nasledovné šifry:

- AES,
- DES,
- Blowfish,
- ARC4.

Príklad použitia symetrickej šifry AES je vyobrazený v nasledujúcom zdrojovom kóde, kde `SecretKeySpec()` vygeneruje špecifikáciu podľa zadaného kľúča (v surovom formáte).

```
SecretKeySpec skeySpec = new SecretKeySpec( (byte[]) key, "AES");
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
byte[] encrypted = cipher.doFinal( (byte[]) data);
```

3.2.3 Asymetrické šifrovanie

Asymetrické šifrovanie využíva vytváranie kľúčových párov. Na generáciu slúži metóda `KeyPairGenerator`, ktorá podľa zvoleného algoritmu vygeneruje kľúčový pár `KeyPair`: súkromný a verejný. Pre asymetrické šifrovanie je možné použiť tieto algoritmy:

- RSA,
- Diffie-Hellman,
- DSA.

Príklad použitia pri algoritme RSA je možné vidieť v nasledujúcom zdrojovom kóde.

```
KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
kpg.initialize(1024); // Nastavi dlzku kluca na 1024b
KeyPair kp = kpg.genKeyPair(); // Vygeneruje klucovy par
```

Kľúče sa po vygenerovaní získajú pomocou metódy `getPrivate()`, pre súkromný kľúč `PrivateKey`, a metódy `getPublic()` pre verejný kľúč `PublicKey`. Rovnakým spôsobom sa generujú páry ďalších podporovaných algoritmov, pričom je možné definovať aj módy a zarovnanie algoritmu. Taktiež je možné vytvoriť vlastné implementácie šifrovacích algoritmov prípadne použiť dostupné riešenia, akým je napr. BouncyCastle.

3.2.4 Hashovacie algoritmy

Jednocestné hashovacie algoritmy sa vytvárajú pomocou `java.security.MessageDigest`. Platforma Android zahŕňa algoritmy:

- MD5,
- SHA1,
- SHA256,
- SHA384,
- SHA512.

Praktickú ukážku použitia algoritmu SHA-1 môžete vidieť na nasledujúcom kóde.

```
MessageDigest md = MessageDigest.getInstance("SHA-1");
md.update(text.getBytes("UTF-8"), 0, text.length());
byte[] sha1hash = md.digest();
```

3.3 Praktické overenie implementácie

Pre praktické overenie modulárnej aritmetikybola vytvorená aplikácia pre platformu Android, ktorá je kompatibilná so zariadeniami s Android 2.2 (API 8) a vyššie.

3.3.1 Časová náročnosť operácií nad veľkými číslami

V tejto aplikácii bola vytvorená aktivita, ktorá vypočíta spotrebovaný čas operácie nad číslami typu `BigInteger`. Na výber okrem zvolenej operácie je možné zadať počet opakovaní (Obr. 3.1). Po spustení testu sa vytvorí osobitné vlákno, ktoré zvolenú operáciu vykonáva

a testuje nad bezpečne generovanými náhodnými číslami pomocou `SecureRandom`, o veľkosti 1024 až 4096 bitov. Prehľad časovej náročnosti základných operácií je vyobrazený v tab. 3.1, ktorý bol testovaný na zariadení HTC Desire. Z tabuľky je zrejmé, že väčšina týchto operácií trvá približne 1 až 3 ms.

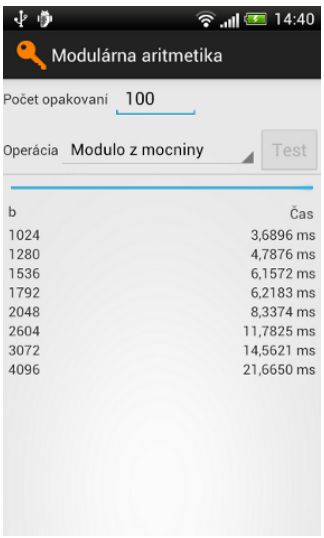
	čas [ms]			
Operácia	**1024 b**	**2048 b**	**3072 b**	**4096 b**
Sčítanie	0,9	1,5	1,7	2,7
Odčítanie	0,9	1,6	1,8	2,7
Násobenie	0,9	1,6	2,2	2,7
Delenie	1,0	1,6	1,8	2,7
Modulo	0,9	1,5	1,9	2,7
10. mocnina	1,4	3,0	4,6	6,7

Tab. 3.1: Časová náročnosť základných operácií čísel `BigInteger`

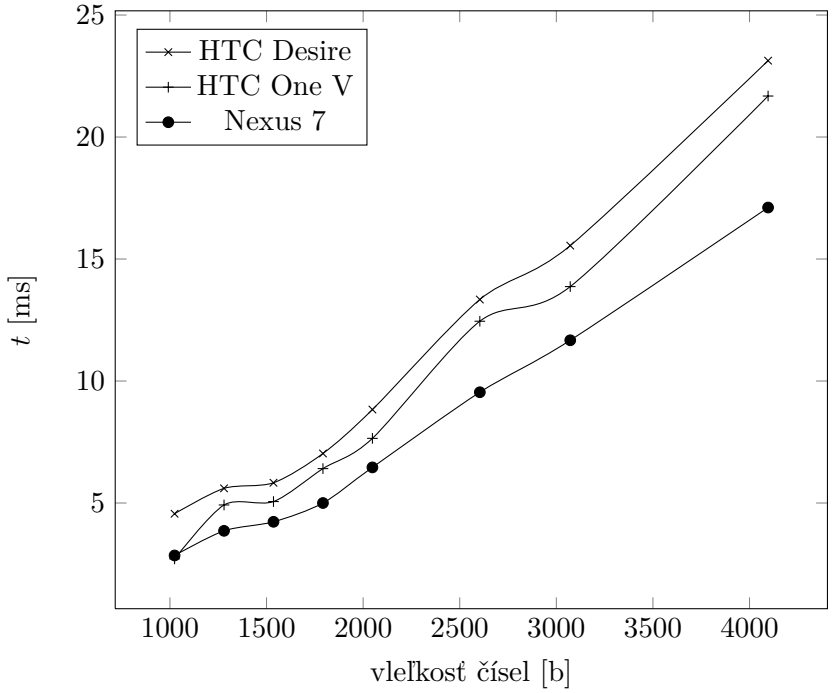
Príklad využitia operácie mocniny s modulom (`modPow`) je možné vidieť pri implementáciách algoritmu RSA, ktorý pracuje s prvočíslami. Pre tento účel bol v aplikácií vytvorený test časovej náročnosti tejto operácie pri 100 opakovaniach za sebou, vyjadrený

$$c^{65537} \mod p \cdot q, \tag{3.2}$$

kde čísla c, p, q sú náhodne generované a majú veľkosť 1024 až 4096 b. Výsledný priebeh je vyobrazený na grafe 3.2, kde na testovanie boli použité telefóny HTC Desire, HTC One V a tablet Nexus 7. Čas potrebný na vykonanie operácie narastal približne lineárne zo zvyšujúcou sa veľkosťou čísel.



Obr. 3.1: Aktivita aplikácie *EncryptionPrimitivesTest* – modulárna aritmetika



Obr. 3.2: Časová náročnosť operácie `modPow` spolu s `multiply` čísel `BigInteger`

4 NÁVRH AUTENTIZAČNÉHO PROTOKOLU

Autentizačný protokol bude využívať chytrý mobilný telefón (smartphone), s ktorého pomocou sa bude užívateľ autentizovať pri prístupe do chránenej služby alebo do chráneného priestoru. Ako typ autentizačného protokolu bola zvolená autentizácia na báze **dôkazu nulovej znalosti** (vid sekcia 1.1.3).

Autentizácia bude prebiehať bezdrôtovo pomocou technológie NFC, a to priložením smartphone-u ku NFC čítačke. Telefón pošle čítačke parametre, ktoré vygeneroval vybraným autentizačným protokolom. Čítačka (server) následne parametre overí, a tým prístup užívateľovi povolí alebo zamietne. Užívateľ je autentizovaný anonymne, takže nemôže byť vystopovatelný. O anonymnej autentizácii pomocou Smart kariet pojednáva dokument [10].

Tento protokol je možné využiť najmä pri prístupových systémoch – do miestností, budov, či serverovní – k prihlasovaniu do elektronických služieb apod.

4.1 Autentizačný protokol HM12

Pre túto prácu bol zvolený autentizačný protokol dôkazu nulovej znalosti **HM12** [11], ktorý bol vytvorený Jánom Hajným s výpomocou Lukáša Malinu. Protokol podporuje revokáciu vylúčených používateľov, ale taktiež aj odhalenie identity, v prípade kritických porušení pravidiel a zákonov. Definuje vytváranie parametrov (registračná fáza), autentizáciu (dôkazová fáza) a revokáciu.

Autentizačný protokol HM12 funguje na princípe rozlišovania reprezentácií diskrétnych logaritmov (DL-REPs). Užívatelia si časť privátnych kľúčov (w_1 a w_2) náhodne volia sami a časť získavajú od revokačného rozhodcu (RR) s pomocou podpisu od vydavateľa. RR vypočíta tajný kľúč w_{RR} s pomocou A_{seed} (bez odhalenia w_1 a w_2) a použitím vlastného tajného kľúča. Privátne kľúče w_1, w_2 a w_{RR} potom tvoria tzv. master kľúč.

V dôkazovej fáze žiadateľ dokazuje overovateľovi (verifikátoru) určitú znalosť (DL-REP), bez samotného odhalenia identity. Bezpečnosť protokolu je založená na probléme nájdenia diskrétneho logaritmu. Overovateľ potom zisťuje, či žiadateľ daný diskrétny log. vie.

Užívateľov je možné odstrániť zo systému odmietaním prístupu, a to:

- odoprením autentizačných parametrov (tokenov) – RR uverejní w_{RR} nechcených užívateľov na čiernu listinu,
- odoprením nespojiteľnosti – RR uverejní w_{RR}, s pomocou ktorého sa spoja predchádzajúce autentizácie užívateľa, bez odhalenia identity,
- odoprením anonymity – s pomocou RR a vydavateľa sa odhalí identita užívateľa.

4.1.1 Autentizačné tokeny

Pred samotnou autentizáciou musí užívateľ nadobudnúť práva, ktoré dosiahne na základe vlastníctva tokenov. Tokeny užívateľ získa v registračnej fáze, kde od autority získa

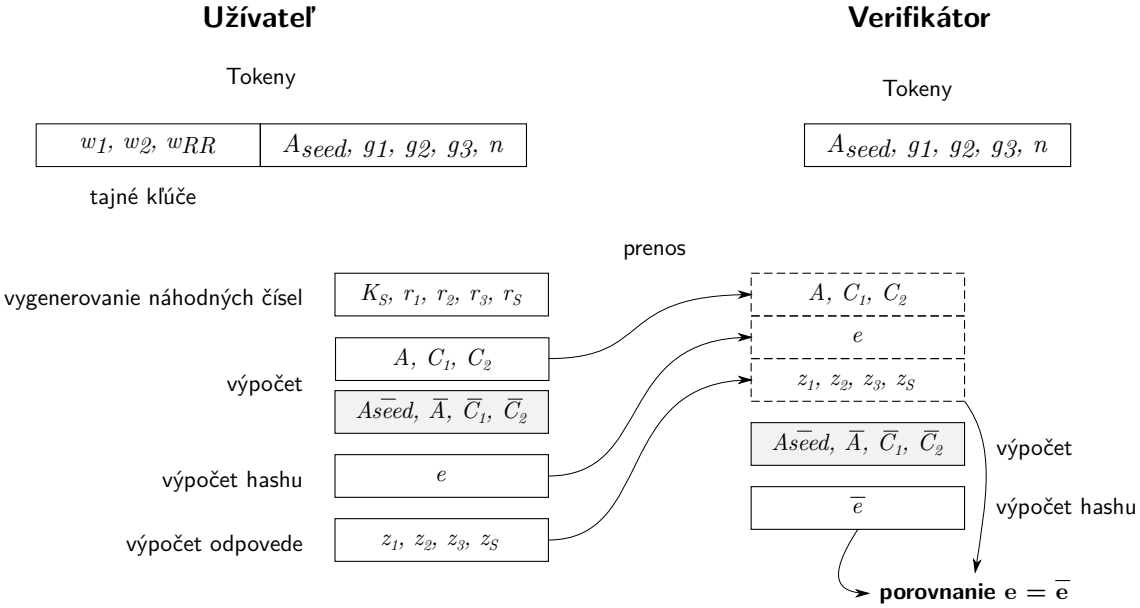
podpísané tajné kľúče (w_1, w_2, w_{RR}) a od autentizačného serveru generačné konštanty $(A_{seed}, g_1, g_2, g_3, n)$. Všetky tokeny sú prvočíslami, pričom ich bitové veľkosti sú uverejnené v Tab. 4.1.

Tab. 4.1: Veľkosti tokenov protokolu HM12

tokeny	**veľkosť [bit]**
w_1	159
w_2	80
w_{RR}	280
A_{seed}, g_1, g_2	1022
g_3	1023
n	1024

4.1.2 Popis autentizácie

Žiadateľ pri autentizovaní vygeneruje náhodné čísla určitých fixných bitových veľkostí $(K_S, r_1, r_2, r_3, r_S)$, ktoré použije pri výpočte diskrétnych logaritmov s využitím tokenov. Následne vypočítané čísla zahashuje, s využitím jednocestného hashu SHA-1, do 160 bitového čísla (e), do ktorého pridá aj údaj o aktuálnom čase v minútach[1]. Potom spolu s hashom vypočíta odpoveď (z_1, z_2, z_3, z_S).



Obr. 4.1: Autentizačný protokol HM12

Verifikátor obdrží určené vypočítané čísla (A, C_1, C_2), vrátane hashu a odpovedi. Na základe týchto čísel dopočíta chýbajúce čísla $(\bar{A_{seed}}, \bar{C_1}, \bar{C_2})$. S pomocou týchto čísel a údajov časových razítok tolerančného okna vypočíta niekoľko hashov, a to rovnakým spôsobom ako na strane užívateľa. Časové tolerančné okno sa volí ± 1 min od aktuálneho času,

[1]Minúty sú vyjadrené ako minúty od 1.1.1970 00:00 UTC

aby bola zaručená správna funkčnosť aj na hraniciach minút, či oneskorenými hodinami v mobilnom telefóne.

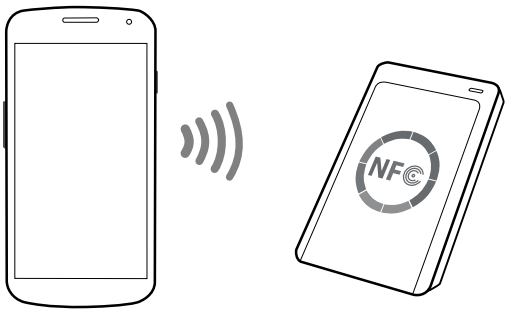
Pri rozhodovaní prístupu sa porovnáva hash prijatý s hashmi vypočítanými (v časovom tolerančnom okne) na strane verifikátoru. V prípade zhody je užívateľovi prístup povolený, v opačnom prípade zamietnutý. Princíp autentizácie je vyobrazený na Obr. 4.1, detailný popis algoritmu HM12 je popísaný v prílohe A.

4.2 Komunikačný prenos NFC

Autentizácia bude prebiehať bezdrôtovo, pomocou technológie NFC. Technológia NFC využíva komunikáciu limitovanú do 10 cm a ako moduláciu používa ASK. Pri rýchlosti kanálu nad 106 kBaud používa kódovaciu schému Manchester. Prenosovú rýchlosť dosahuje max. 424 kbit/s.

Použitá bude technológia NFC-A, v ktorej sa dáta štandardného rámca skladajú z payloadu (obsahuje príkaz alebo odpoveď) a EoD (End of Data). EoD obsahuje dvoj-bajtový CRC checksum, vypočítaný z payloadu. Dáta rámca sú ďalej opatrené paritnými bitmi nepárnej parity, za každým blokom ôsmych bitov (oktet). [12]

Vypočítané parametre autentizačného protokolu budú posielané jednosmerne z mobilného telefónu do čítacieho zariadenia. Pre výmenu správ bol využitý Android Beam™, ktorý je založený na protokole SNEP.



Obr. 4.2: Ilustrácia NFC (telefón–čítačka)

4.2.1 NDEF správy

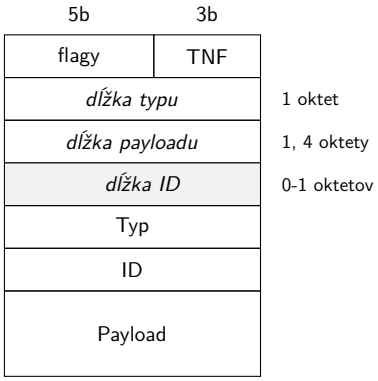
NDEF správy (messages), posielané protokolom SNEP, obsahujú jeden alebo viac NDEF záznamov (records). NDEF záznamy obsahujú tri parametre pre špecifikáciu payloadu (dát), a to: dĺžku, typ a identifikáciu payload-u. Payload záznamu môže dosahovať dĺžku až $2^{32} - 1$ bajtov ($\sim$4 GiB), v prípade väčšej dĺžky musia byť záznamy rozdelené na časti, tzv. chunky.

Štruktúra NDEF záznamu začína flagmi, ktoré definujú začiatky a konce správy, prípadne použitie krátkej správy, či použitia políčka ID. Typový názov formátu (TNF) špecifikuje význam políčka typu záznamu. Používané TNF sú napr.:

- *Well-known type* – známy typ, ďalej špecifikovaný v políčku typ záznamu,

- *Media type* – médium, MIME uvedené v type,
- *URI* – absolútna URI adresa,
- *Unknown* – neznámy formát.

Do políčka typu záznamu sa potom uvádza konkrétny typ payloadu, napr. pri použití TNF *Well-known type* môže byť typom payloadu RTD_TEXT (textový reťazec). Políčko ID sa používa pre identifikáciu záznamu, pričom musí byť nastavený flag použitia tohto políčka (IL). [13]



Obr. 4.3: Štruktúra NDEF záznamu

4.2.2 Prenos čísel autentizačného protokolu

Užívateľská aplikácia, po priložení mobilného telefónu ku čítaciemu zariadeniu, vygeneruje autentizačné parametre (čísla), pomocou protokolu HM12 (4.1). Vypočítané parametre $A, C_1, C_2, e, z_1, z_2, z_3$ a z_S pošle vo forme NDEF správy, kde každý parameter vystupuje ako NDEF záznam.

Záznamy sú identifikované pomocou ID, ktoré začína od ID 0, pre parameter A, a končí ID 7, pre‘parameter z_S – v rovnakom poradí sú aj zabalené v NDEF správe. Posielajú sa s typom TNF *Unknown*, kde payload záznamu nesie binárne dáta daných čísel variabilných veľkostí.

5 IMPLEMENTÁCIA AUTENTIZAČNÉHO PROTOKOLU

Autentizačný protokol HM12 bol implementovaný do užívateľskej Android aplikácie a do serverovej aplikácie *beam-server*, spustenej na osobnom počítači s NFC čítačkou.

5.1 Android aplikácia (klient)

Vytvorená Android aplikácia *NfcAuthZK* využíva technológiu Android Beam™, založenú na posielaní správ pomocou NFC. Vyžaduje verziu Android 4.0 (SDK 14) a vyššiu, v ktorej bola NFC komunikácia prvotne zavedená.

5.1.1 Stavba aplikácie

Aplikácia obsahuje dve základné aktivity: `NfcAuthActivity` a `SecretTokenSetActivity`.

Aktivita `NfcAuthActivity` je hlavnou aktivitou aplikácie – spúšťa sa ako prvá. Pri jej vytvorení sa získa primárny NFC adaptér smartphone-u a vytvorí sa objekt spätných volaní, ktorý vytvorí objekt autentizačného protokolu vrátane jeho parametrov. Spätné volania NFC sa pridelia adaptéru a sú volané, buď pred samotným prenosom (priložením k čítačke) alebo po kompletnom odoslaní správy, tak ako znázorňuje nasledujúci kód.

```
NfcAdapter mNfcAdapter = NfcAdapter.getDefaultAdapter(this);
NfcAuthCallbacks nfcCallbacks = new NfcAuthCallbacks(this);

// registrácia spätného volania spúšťaného pred prenosom
mNfcAdapter.setNdefPushMessageCallback(nfcCallbacks, this);
// registrácia spätného volania spúšťaného po kompletnom odoslaní
mNfcAdapter.setOnNdefPushCompleteCallback(nfcCallbacks, this);
```

Vytvorením objekt `NfcAuthCallbacks` sa vytvorí aj objekt autentizačného protokolu vrátane parametrov, ktoré získajú zo zdieľaného úložiska nastavení a z úložiska Credential Storage.

Aktivita `SecretTokenSetActivity` slúži na nastavenie parametrov (tokenov) autentizácie a je možné ju vyvolať pomocou MENU > *Settings* > *Set Tokens*. Tokeny sa sťahujú pomocou protokolu HTTPS, z URL adresy zadanej do textového poľa. Ako formát bol použitý univerzálny jazyk XML, ktorého forma je uvedená v prílohe B. Užívateľ je o stave stiahnutia dát informovaný pomocou vyskakujúcej správy, tzv. Toast-u.

5.1.2 Uloženie tokenov

Tokeny sa skladajú z generačných konštánt a tajných kľúčov a sú vyjadrené ako čísla `BigInteger`. Vo forme textových reťazcov sa vyjadrujú v desiatkovej sústave, a tak sú aj načítavané zo súboru XML.

Generačné konštanty sa ukladajú do úložiska zdieľaných nastavení aplikácie. Do úložiska sa ukladajú ako textové reťazce a rozlišujú sa podľa názvov (kľúčov), uvedených v prílohe B.

```
SharedPreferences sharedPref = PreferenceManager.
    getDefaultSharedPreferences(this);

Editor prefEdit = sharedPref.edit();
prefEdit.putString((String) key, (String) value); // uloženie hodnoty
sharedPref.getString((String) key, "0"); // načítanie hodnoty, 0 ak hodnota
     nebola nastavená
```

Tajné kľúče sú ukladané do zabezpečeného úložiska Credential Storage (viď 2.2.2). Zabezpečené úložisko je nutné pred jeho používaním odomknúť, o čo sa stará trieda `SecretKeyStore`. Používať je ho možné len v prípade, ak je nastavené jeho heslo alebo je telefón zabezpečený PIN kódom, heslom alebo gestom.

Credential Storage funguje na princípe lokálnych socketov, kde je toto úložisko dostupné na adrese UNIX-ovej domény „keystore“. Tajné kľúče sa ukladajú ako textové reťazce vo forme bajtov, identifikované pomocou mena podobne ako tokeny generačných konštánt. Príklad použitia uvádza nasledujúci kód.

```
// Odomknutie a pripojenie do úložiska
SecretKeyStore ks = new SecretKeyStore((Context) this);

ks.put((String) key, (byte[]) byteValue); // uloženie tajného kľúča
(byte[]) ks.get((String) key); // načítanie kľúča
```

V rámci demonštrácie a funkčnosti sa autentizačné parametre sťahujú z webového serveru a až potom ukladajú do secure elementu. Každopádne pri plnohodnotnom nasadení autentizačného protokolu HM12 je potrebné parametre získa vo fáze registrácie (tzv. `IssueAtt` protokol) viď [11]. Súbor XML s testovacími autentizačnými parametrami je uverejnený v prílohe C.[1]

5.1.3 Autentizácia a prenos parametrov

Pre autentizačný protokol bola vytvorená trieda `HM12`, ktorá obsahuje všetky procedúry pre generovanie odpovede a hashu. Počas inicializácie sa nastavia generačné konštanty A_{seed}, g_1, g_2, g_3 a n, načítané zo zdieľaného úložiska nastavení.

Pre matematické operácie sú využívané metódy triedy `BigInteger`, a to `multiply()` pre násobenie, `mod()` pre modulo a `modPow()` pre umocnenie s modulom. Náhodné čísla sa generujú pomocou rovnakej triedy, kde sa do konštruktora uvedie bitová dĺžka a objekt pseudo-náhodných čísel. Keďže trieda generuje čísla 0 až $2^n - 1$ (n je bitová dĺžka), je nutné zaručiť presnú bitovú dĺžku čísla, ktorú vyžaduje autentizačný protokol. Bitová dĺžka je zaručená nastavením $(n-1)$-tého bitu.

```
BigInteger num = new BigInteger((int) numBits, (Random) rnd);
```

[1] Alebo je možné použiť ako URL `https://www.stud.feec.vutbr.cz/~xkriza03/auth/tokens.xml`.

```
num = num.setBit(numBits -1); // zaručenie bitovej dĺžky
```

Hash sa generuje pomocou SHA-1, do ktorého sa vkladajú bajtové polia (`byte[]`) čísel A, $\bar{A}$, A_{seed}, $\bar{A_{seed}}$, C_1, C_2, $\bar{C_1}$ a $\bar{C_2}$ vypočítaných algoritmom. Ďalej je do hashu vložené bajtové pole o aktuálnych minútach[2] v základnom časovom pásme GMT. Bajtový výstup hashu je vstupom pre novo vytvorené číslo `BigInteger`, ktoré môže nadobúdať záporné hodnoty, a preto sa berie jeho absolútna hodnota. Číslu je taktiež nastavený MSB bit, aby bola zaručená jeho bitová dĺžka.

Prenos parametrov

Autentizačný proces začína priložením smartphone-u k čítaciemu zariadeniu, ktoré má spustený program *beam-server*. Na telefóne musí byť zapnutá aplikácia a jej hlavná aktivita.

Po úspešnej inicializácii telefón–čítačka, sú na telefóne splnené tieto kroky:

1. z Credential Storage sú načítané tajné kľúče (w_1, w_2, w_{RR}),
2. algoritmom HM12 je vypočítaná odpoveď z načítaných a uložených tokenov,
3. z vypočítanej odpovede je vytvorená NDEF správa, kde každý parameter je zabalený do NDEF záznamu,
4. NDEF správa je po potvrdení užívateľom poslaná čítačke pomocou systémového API a NFC adaptéru.

NDEF správa je zložená z niekoľkých NDEF záznamov, ktorých payload obsahuje bajtové hodnoty z vypočítaných čísel `BigInteger` odpovede. Každý NDEF záznam je identifikovaný pomocou ID a ako jeho typový názov formátu bol zvolený *Unknown* (viď. 4.2.1). Správa je vytvorená ešte pred samotným prenosom, t.j. po inicializácii spojenia s čítačkou.

Tab. 5.1: Identifikácia NDEF záznamov a ich premenných

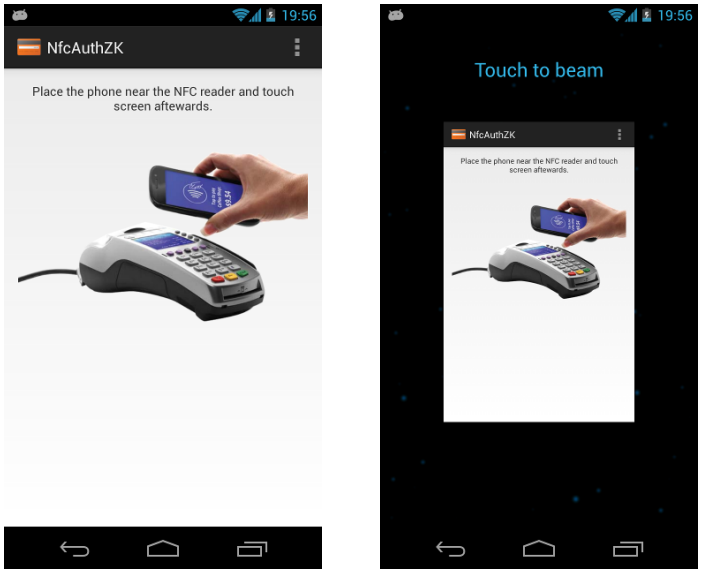
ID záznamu	**Kľúč**	**Premenná**
0	`A`	A
1	`C1`	C_1
2	`C2`	C_2
3	`e`	e
4	`z1`	z_1
5	`z2`	z_2
6	`z3`	z_3
7	`zs`	z_S

Po vygenerovaní odpovede a vytvorení NDEF správy je užívateľ vyzvaný k potvrdeniu odoslania správy NFC adaptérom – na displeji sa zobrazí hlásenie „Touch to beam“ (Obr. 5.1b). Počas hlásenia môže užívateľ potvrdiť odoslanie dotykom obrazovky (v oblasti

[2]Minúty sú vyjadrené ako minúty od 1.1.1970 00:00 UTC

aplikácie), ktoré sa zobrazuje len po dobu spojenia s čítačkou. Po potvrdení a kompletnom odoslaní správy je užívateľ informovaný pomocou Toast správy. Odoslanie je možné zamietnuť buď oddialením telefónu od čítačky alebo stlačením tlačidla späť.

Pre správnu funkčnosť aplikácie je nutné zapnúť funkciu Beam v nastaveniach telefónu, *Settings* > *Wireless & Networks – More...* > *Android Beam* > ON.



(a) (b)

Obr. 5.1: Android aplikácia s implementovaným autentizačným protokolom

5.2 Serverová aplikácia *beam-server*

Na strane overovateľa (verifikátora) bola vytvorená serverová aplikácia *beam-server*, ktorá je spustená na osobnom počítači. Aplikácia je napísaná v jazyku JAVA a využíva *Smart Card I/O API* (`javax.smartcardio`).

Ako NFC čítacie zariadenie bolo použité ACS ACR122U (Obr. 5.2), napájané a pripojené pomocou USB. Ovládače sú dostupné na priloženom DVD alebo na stránke výrobcu[3].

5.2.1 Stavba aplikácie

Serverová aplikácia je založená na *Java Android Beam API*[4] (ďalej len Beam API), ktorého autorom je Wilko Oley. Použitá verzia Beam API neobsahovala podporu viacerých NDEF záznamov v jednej správe. Túto funkcionalitu sme implementovali, a to s pomocou rozdeľovania surových dát na základe analyzovaných bajtových veľkostí záznamov.

[3] `http://www.acs.com.hk/index.php?pid=product&id=ACR122U`
[4] `http://code.google.com/p/java-android-beam-api/`

Obr. 5.2: NFC čítačka ACR122U

Hlavnou triedou aplikácie je `Server`, ktorá po vytvorení inicializuje triedu autentizačného protokolu `HM12` vrátane parametrov (generačných konštánt). Ďalej detekuje pripojenú NFC čítačku a nastaví spätné volanie spúšťané po úspešnom prijatí Beam (NDEF) správy.

Ovládač Beam API `ACR122U` vyberie prvé dostupné čítacie zariadenie (terminál) a prejde do stavu čakania na spojenie s mobilným telefónom (alebo smart kartou). Po priložení mobilného telefónu vytvorí komunikačný kanál, využívajúci aktuálne dostupný protokol. Následne telefón vyzve k spojeniu na logickej úrovni – tzv. beam handshake, a začne prijímať odoslanú NDEF správu, ktorú rozoberie na NDEF záznamy a predá spätnému volaniu.

```
TerminalFactory factory = TerminalFactory.getDefault();
List<CardTerminal> list = factory.terminals().list(); // zoznam čítačiek
terminal = list.get(0); // výber prvej čítačky

terminal.waitForCardPresent(0); // čakanie na telefón/kartu
(Card) card = terminal.connect("*"); // vytvorenie kanálu

// Čakanie na Android Beam a prijatie správy
// AcsNFCDevice.whaitForAndroidBeam()
NdefMessage message = whaitForAndroidBeam(timeout, max_allowed_size);
```

Do Beam API bola implementovaná aj nekonečná slučka, ktorá opakovane vykonáva procedúry čakania na telefón a prijímania správy, aj po jej úspešnom prijatí. Na ďalšiu správu čaká približne po 1,5 s.

5.2.2 Závislosti a spustenie aplikácie

Samotná aplikácia vyžaduje pre jej kompiláciu tieto JAVA knižnice:

- Simple Logging Facade for Java (or SLF4J)
- Apache log4j™
- Apache Commons Codec™
- Apache Commons Lang

Všetky tieto balíčky sú zabalené do spustiteľného súboru JAR aplikácie alebo jednotlivo na priloženom DVD.

Smart Card I/O vyžaduje na OS Linux program *PCSC Lite* a jeho knižnice. V niektorých distribúciách Linuxu je nutné uviesť, pred spustením aplikácie, cestu k PCSC knižnici, a to ako parameter pre JAVA VM (systémová premenná `_JAVA_OPTIONS`):

```
-Dsun.security.smartcardio.library=/usr/lib/x86_64-linux-gnu/libpcsclite.so
```

Aplikácia sa spúšťa z príkazového riadku operačného systému, keďže ide o konzolovú aplikáciu. Všetky hlásenia sa zobrazujú na štandardnom výstupe.

```
java -jar beam-server.jar [blacklistfile]
```

`blacklistfile` je textovým súborom čiernej listiny (napr. blacklist.txt).

Pre OS Linux bol vytvorený spustiteľný shell script, ktorý v prípade chyby aplikáciu po 4 s reštartuje. Spúšťa sa

```
./run-beam-server.sh [blacklistfile]
```

5.2.3 Príjem parametrov a autentizácia

Aplikácia po spustení čaká na priloženie mobilného telefónu (alebo smart karty). Z mobilného telefónu sa s pomocou protokolu SNEP prijíma NDEF správa. Správa obsahuje NDEF záznamy, ktoré reprezentujú jednotlivé autentizačné premenné. Payload záznamov je vo forme bajtov, z ktorých sa vytvoria čísla typu `BigInteger`.

Program po priložení telefónu vykoná tieto kroky:

1. vytvorí sa komunikačný kanál medzi čítačkou a telefónom,
2. vyšle sa požiadavka na spojenie – Beam handshake,
3. po úspešnom spojení začne čítačka prijímať NDEF správu odoslanú z telefónu pomocou SNEP protokolu,
4. z NDEF správy sa vyberú NDEF záznamy, ktoré reprezentujú autentizačné premenné, a zaradia sa do premenných programu,
5. správnosť premenných je vyhodnotená autentizačným protokolom HM12

Autentizácia

Autentizačný protokol bol implementovaný do triedy `HM12`, podobne ako v prípade Android aplikácie (viď 5.1.3).

Hlavnou metódou je `proveAttCheck()`, ktorá vykonáva výpočty autentizácie nad číslami typu `BigInteger`, na základe uvedených argumentov. Jej súčasťou je výpočet kontrolného hashu, do ktorého sa okrem autentizačných premenných pripája aj časové razítko vo forme minút. Kontrolný hash sa porovná s hashom prijatým (e) od užívateľa, a na základe zhody je užívateľ úspešne autentizovaný.

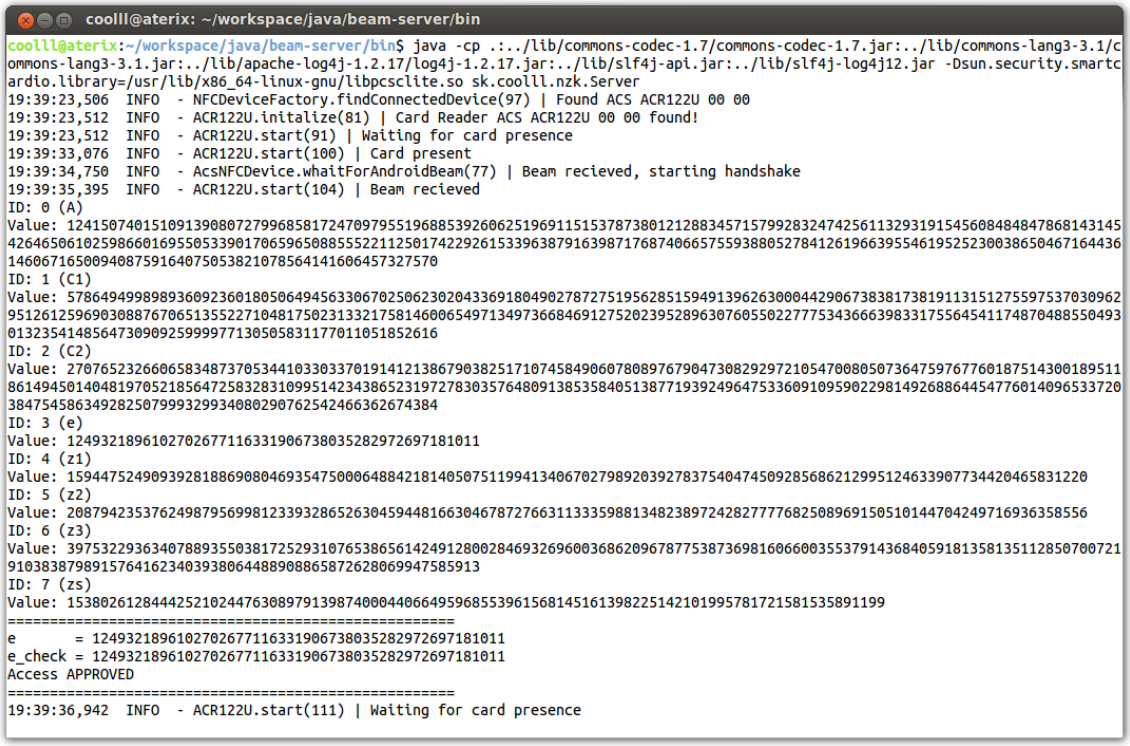
Kvôli oneskoreniam na prenose autentizačných parametrov alebo zlou synchronizáciou systémového času bolo zavedené časové tolerančné okno. Tolerančné okno sa vytvára z oboch strán aktuálneho času – zvolené je okno ± 1 min. Pre tento účel sa používa metóda `proveAttCheckTolerant()`, ktorá prijíma ako argument jeho veľkosť, a vykonáva kontroly hashov práve v tomto okne.

Nechcený užívatelia sú uvádzaní na čiernu listinu, a to odoprením autentizačných parametrov. Čierna listina sa načíta z textového súboru, kde sú odoprené w_{RR} oddelené novým riadkom. Metóda `isBlacklisted()` overí, či sa daný užívateľ nenachádza na listine, a to s pomocou vzorca:

$$C_1 \stackrel{?}{=} C_2^{rev} \mod n, \tag{5.1}$$

kde $rev = w_{RR}$.

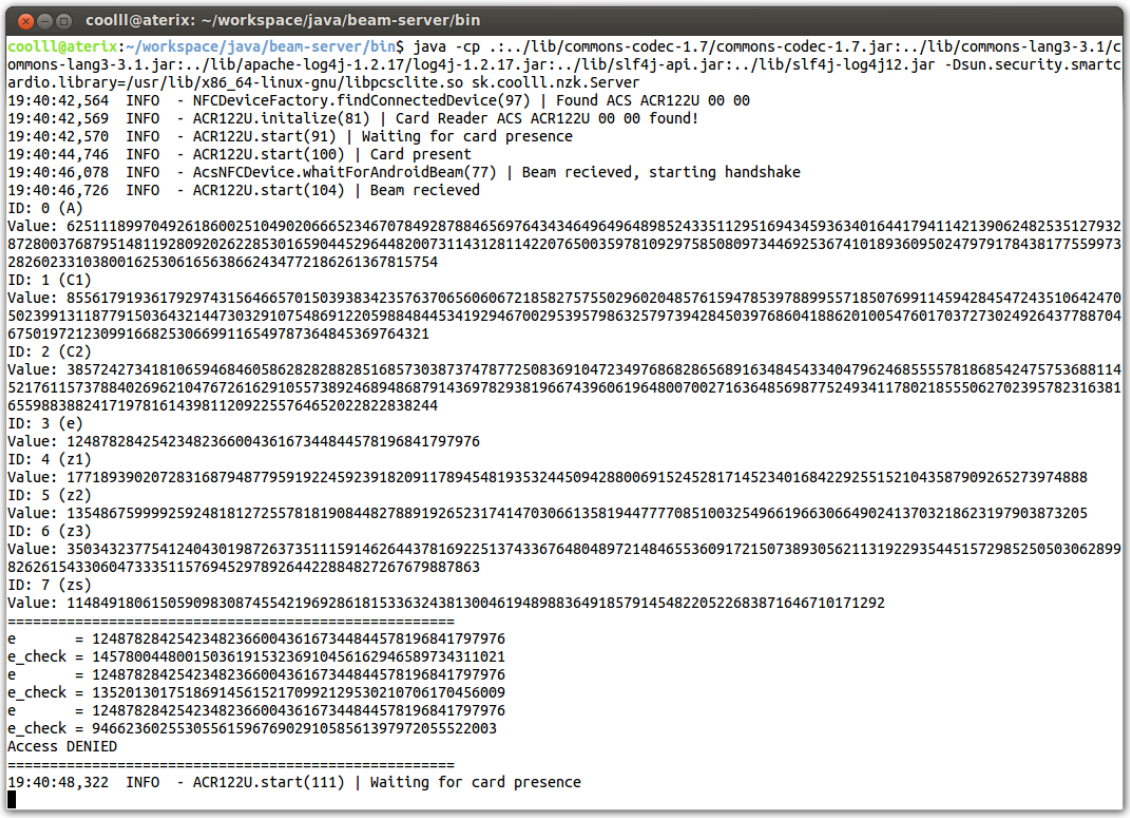
Praktická ukážka funkčnosti autentizácie je zobrazená na obr. 5.3, na ktorom je možné vidieť prijatie autentizačných premenných (A, C_1, C_2, e, z_1, z_2, z_3, z_S) od užívateľa, identifikovaných podľa tab. 5.1. Taktiež je zrejmá zhodnosť hashu prijatého (`e`) a hashu kontrolného (`e_check`), a tým je užívateľ úspešne autentizovaný (*„Access APPROVED“*).



Obr. 5.3: Serverová aplikácia *beam-server* – úspešná autentizácia

Príklad neúspešnej autentizácie je zobrazený na obr. 5.4, kde je možné vidieť trojnásobnú kontrolu hashov, v rámci tolerančného okna ± 1 min. Prvý kontrolný hash bol vypočítaný v čase t, druhý v čase $t-1$ a tretí v čase $t+1$, kde t je aktuálny čas v minútach. Ani jeden z kontrolných hashov sa nezhodoval z hashom prijatým, a preto bol užívateľovi prístup zamietnutý (*„Access DENIED“*). V prípade, ak je užívateľ uverejnený na čiernej

listine, je autentizácia taktiež neúspešná – *„Access DENIED – blacklisted“*



Obr. 5.4: Serverová aplikácia *beam-server* – neúspešná autentizácia

6 TESTOVANIE A ZHODNOTENIE BEZPEČNOSTI IMPLEMENTÁCIE

Android aplikácia *NfcAuthZK* bola primárne testovaná na telefóne *Galaxy Nexus*, ďalej na *Samsung Galaxy S3* a na tablete *Nexus 7*. Serverová aplikácia *beam-server* bola spustená na operačnom systéme Ubuntu 13.04 (amd64) s Java SE Runtime Environment verzie 1.7. Ako NFC čítačka bola použitá ACS ACR122U-A2 (Obr. 5.2), pripojená pomocou USB.

Pre operačný systém Microsoft Windows sa serverovú aplikáciu *beam-server* nepodarilo úspešne spustiť. Dôvodom bolo *Java Android Beam API*, ktoré v dobe písania práce nebolo v stabilnej verzii, a taktiež skutočnosť, že telefóny pri kontakte s NFC poľom vysielajú inicializačnú správu prerušovane – spojenie udržujú až po úspešnom nadviazaní spojenia.

Aplikáciu je možné použiť na serveroch alebo na embedded systémoch s OS Linux. Pre účely testovania bol vytvorený virtuálny operačný systém bežiaci vo VirtualBox-e, ktorý sa nachádza na priloženom DVD (príloha C).

6.1 Bezpečnosť navrhnutého protokolu

Útok autentizačného protokolu spočíva v tom, že si užívatelia môžu vytvárať veľké množstvo master kľúčov, ktoré súhlasia s A_{seed}.

$$A_{seed} = g_1^{w_1+\alpha(w_1-\hat{w}_1)} g_2^{w_2+\alpha(w_2-\hat{w}_2)} g_3^{w_{RR}+\alpha(w_{RR}-\hat{w}_{RR})}, \tag{6.1}$$

kde $\hat{w}_1, \hat{w}_2$ a $\hat{w}_{RR}$ sú novo zvolené kľúče a kde $\alpha = q - 1$. Tým vytvoria nové a platné parametre, kde exponent g_3 môže byť zvolený náhodne, a to taký, ktorý nie je na čiernej listine systému. Tomuto útoku je možné zabrániť použitím určitej sady reprezentácií diskrétnych logaritmov (DL-REP) pevných hodnôt, ktoré by mali zahŕňať nelineárne vzťahy medzi exponentmi. [15]

Navrhnutý protokol používa jednosmerný prenos parametrov, užívateľ teda nereaguje na výzvu, s pomocou ktorej vytvorí odpoveď. Odpoveď sa generuje len na strane užívateľa. Práve preto je do kontrolného hashu pridávaná časová informácia, aby nedošlo k použitiu parametrov, odchytených útočníkom, v iný čas, než je veľkosť časového okna. Môže dôjsť k situácii, kde útočník, po úspešnom odblokovaní telefónu a credential storage, nazbiera autentizačné parametre v určitých časoch. Útočník si čas nastaví v nastaveniach telefónu a nazbierané parametre použije v presne daných okamihoch.

6.2 Využitie bezpečnostného elementu

Android aplikácia využíva na uloženie tajných kľúčov zabezpečené úložisko Credential Storage (sekcia 2.2.2). Do úložiska sa tajné kľúče ukladajú pre každú aplikáciu zvlášť a sú prístupné až po jeho odomknutí. Odomknutie vykonáva užívateľ, a to buď zadaním hesla

úložiska alebo odomknutím telefónu s pomocou gesta, hesla alebo PIN kódu. Bezpečnosť aplikácie potom závisí na zvolenom hesle, PIN kóde alebo geste.

Aplikácia neobsahuje dodatočnú autentizáciu pre jej spustenie a používanie. Využíva však výhodu uzavretého zabezpečeného úložiska, ktoré ak zostane pre útočníka uzavreté, tak nedôjde k úspešnému načítaniu tajných kľúčov, a tým ani k úspešnej autentizácii.

Tajné kľúče, uložené v Credential Storage, je možné prečítať z pamäťového priestoru aplikácie (v RAM), keďže sa načítavajú do premenných programu. Pre úspešné vykonanie tohto úkonu je nutný predpoklad odomknutia zabezpečeného úložiska a taktiež dostupnosť ladiacich nástrojov Android-u (*Dalvik Debug Monitor*).

6.3 Zhodnotenie NFC komunikácie

NFC komunikácia je zabezpečená kanálovým kódovaním – nepárnou paritou a CRC checksum-om. NDEF správy sú fragmentované do niekoľkých fragmentov, pričom výsledná správa dosahuje veľkosť okolo **664 Byte**.

Testované mobilné telefóny využívali technológiu `RFID - ISO 14443 Type A Part 3`. Vysielané RFID bolo vždy variabilné, a tak nie je možná spojiteľnosť s konkrétnym zariadením a užívateľom. Tým je zachovaná anonymita užívateľa.

Útokom na NFC komunikáciu môže byť tzv. útok typu „Man in the Middle“. Keďže prenos NDEF správ cez NFC je nezabezpečený, užívateľ nevie overiť, či sa jedná o správny alebo podvrhnutý verifikátor (čítačku). Užívateľ môže prísť k podvrhnutej čítačke, priložiť telefón a následne odoslať autentizačné parametere, ktoré sú potom podvrhnutým verifikátorom použité v rovnakom čase, ale na inom mieste. Potom už záleží na rýchlosti prenosu parametrov z podvrhnutej čítačky cez prenosový kanál útočníkovi. Podobne je možné vykonať aj odpočúvanie prenosu v blízkosti správnej čítačky. Vyriešiť tento problém je možné pomocou autentizácie a overenia komunikácie so správnym čítacím terminálom, a to napr. použitím certifikátov RSA alebo ECDSA. Viac o bezpečnosti NFC komunikácie diskutuje [16].

7 ZÁVER

Mobilný telefón poskytuje množstvo riešení pre autentizáciu užívateľa voči telefónu a autentizáciu užívateľa s použitím mobilného telefónu ako tokenu. Vďaka jeho hardvérovému vybaveniu a prístupu na internet, s čoraz väčšími rýchlosťami prenosu dát, je možné vytvoriť zložitejšie a bezpečnejšie metódy autentizácie.

Pre praktické overenie modulárnej aritmetiky kryptografických primitívov bola vytvorená aplikácia pre platformu Android, ktorá merala časovú náročnosť základných operácií nad číslami typu `BigInteger`. Testované boli náhodné čísla s veľkosťami 1024 až 4096 b, pričom ich operácie trvali približne od 1 do 3 ms. Pri použití väčších čísel a zvýšenej bezpečnosti sa zvyšuje aj náročnosť na výpočet, čo súvisí aj s dlhším časom, potrebným na šifrovanie a dešifrovanie. Na platforme Android sú dostupné symetrické šifry AES, DES a ARC4, asymetrické šifry RSA, DH a DSA, a jednosmerné hashovacie funkcie MD5, SHA1, SHA256.

Pre autentizáciu pomocou mobilného telefónu sme zvolili protokol nulovej znalosti HM12 [11], ktorý je možné využiť pri prístupoch do budov alebo pri požičiavaní kníh z knižníc. Tokeny (autentizačné parametre) tvoria generačné konštanty A_{seed}, g_1, g_2, g_3, n a tajné kľúče w_1, w_2, w_{RR} (master kľúč). Pri autentizácii užívateľ vystupuje anonymne, kde overovateľovi dokazuje, že pozná určitú znalosť – diskrétny logaritmus. Užívateľov je možné zo systému vylúčiť odoprením tokenov, odoprením nespojiteľnosti (s minulosťou) alebo aj odoprením anonymity (v kritických udalostiach).

Autentizačné premenné sú posielané jednosmerne, bez spätnej väzby, pomocou technológie NFC z mobilného telefónu do čítacieho zariadenia. Zabalujú sa do NDEF záznamov a tvoria NDEF správu, ktorej veľkosť je ∼664 Byte. Na strane overovateľa sa porovnáva hash prijatý s hashom vytvoreným, z parametrov prijatých a vypočítaných. NFC technológia pre linkové zabezpečenie využíva paritné bity a CRC checksum, dosahuje max. prenosovú rýchlosť 425 kbit/s.

Autentizačný protokol sme implementovali do užívateľskej aplikácie *NfcAuthZK* platformy Android. Aplikácia využíva NFC dispečera systému (Background Tag Dispatch System) pre prenos NDEF správ, ktoré prenáša s typom *Unknown*, pretože sa jedná o nešpecifikovaný tag (štandard). Všetky potrebné tokeny si pre testovacie účely užívateľ sťahuje z webového serveru s využitím šifrovaného spojenia (protokol HTTPS). Generačné konštanty autentizácie aplikácia ukladá do zdielaných nastavení. Master kľúče autentizácie ukladá do zabezpečeného úložiska Credential Sorage, kde uložené dáta sú dostupné len pre danú aplikáciu, a to až po odomknutí úložiska užívateľom. V prípade, ak útočník získa dáta (súbory) z Credential Storage, tak ich bez master kľúča úložiska nedešifruje, keďže sú šifrované 128-bitovým AES. Tokeny a autentizačné premenné sú vyjadrované ako veľké celé čísla typu `BigInteger`, s ktorými je vykonávaná modulárna aritmetika.

Pre overovanie parametrov a samotnú autentizáciu sme vytvorili aplikáciu *beam-server*, ktorá prijíma autentizačné premenné od užívateľa a rozhoduje o úspešnosti autentizácie. Aplikácia je spustená primárne na Linuxových operačných systémoch, ktoré majú pri-

pojenú NFC čítačku. Využíva *Smart Card I/O API* jazyku Java a je založená na *Java Android Beam API*, do ktorého sme implementovali podporu prijímania viacerých záznamov v jednej správe. Aplikácia sa spúšťa cez príkazový riadok systému. Základ aplikácie tvorí dôkazová fáza autentizačného protokolu, kde sa do hashov pridáva aj časové razítko (aj na strane užívateľa). Časové razítko sa pridáva kvôli použiteľnosti vypočítaných autentizačných premenných len v čase výpočtu, s prihliadnutím na toleranciu (časové tolerančné okno), keďže sa jedná o jednosmerný prenos.

Užívateľská aplikácia je chránená uzamknutým Credential Storage, bez ktorého nie je možné sa úspešne autentizovať. Jej bezpečnosť závisí na sile master kľúča zabezpečeného úložiska, ktoré je derivované z hesla, gesta alebo PIN kódu mobilného telefónu. Autentizačný protokol obsahuje bezpečnostný nedostatok, ktorý dovoľuje užívateľom vytvárať nové a zároveň platné autentizačné parametre podľa určitých pravidiel. Predpokladom pre tento typ útoku je znalosť parametrov master kľúča autentizácie. Dôvodom vzniknutia tohto nedostatku môže byť skutočnosť, že autentizačný protokol HM12 je určený najmä na Smart karty, do ktorých nahráva autentizačné parametre určitá autorita, a ktorých pamäť je len ťažko čitateľná.

Keďže prenos autentizačných premenných medzi mobilným telefónom a čítacím zariadením prebieha jednosmerne, tak nie je možné informovať užívateľa o úspešnosti autentizácie. Užívateľa je možné informovať o úspešnosti napr. svetelným signálom (zelená LED). Obojsmerný prenos nepodporovalo, v dobe písania práce, *Java Android Beam API*, ktoré ale bolo výhodnou voľbou, z dôvodu podpory fragmentácie správ.

Serverová aplikácia môže byť spustená na embedded systémoch, prípadne na serveroch, kde sú čítacie zariadenia pripojené cez zbernicu. Výrobca použitej čítačky ACR122U poskytuje aj ovládač pre systém Android, a tak je možné toto zariadenie pripojiť aj na systémy bežiace na tejto platforme. Autentizačný protokol využívajúci technológiu NFC pre prenos premenných si nájde uplatnenie v prístupových systémoch, akými sú budovy firiem, či škôl, v knižniciach pri požičiavaní kníh, a pod.

LITERATÚRA

[1] BURDA K. *Bezpečnost informačních systémů.* Brno: 2005.

[2] STALLINGS W. *Cryptography and Network Security.* 4th ed. Upper Saddle River: Pearson Prentice Hall, 2006, 592 s. ISBN 01-318-7316-4.

[3] MENEZES A., VAN OORSCHOT P., VANSTONE S. *Handbook of applied cryptography.* Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7.

[4] ETSI TS 100 977. *Digital cellular telecommunications system (Phase 2+): Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface.* Sophia Antipolis Cedex, FRANCE, 2007. Dostupné z: <`http://www.3gpp.org/ftp/specs/html-info/1111.htm`>.

[5] KHAN W., ULLAH H. Authentication and Secure Communication in GSM, GPRS, and UMTS Using Asymmetric Cryptography. *International Journal of Computer Science Issues.* 2010, Issue 3, No 9. ISSN 1694-0784.

[6] Storing application secrets in Android's credential storage. ELENKOV, N. *Android Explorations* [online]. 4.6.2012 [cit. 2012-11-26]. Dostupné z: <`http://nelenkov.blogspot.sk/2012/05/storing-application-secrets-in-androids.html`>

[7] PSUTKA J., MÜLLER L., MATOUŠEK J., RADOVÁ V. *Mluvíme s počítačem česky.* Vyd. 1. Praha: Academia, 2006, 746 s. ISBN 80-200-1309-1.

[8] LI S., JAIN A. K. *Handbook of face recognition.* New York: Springer Science Business Media, 2004, 395 s. ISBN 978-0-387-40595-7.

[9] MEDNIEKS Z., DORNIN L., MEIKE G. B. *Programming Android.* 2nd ed. Farnham: O'Reilly, 2012, 524 s. ISBN 978-144-9316-648.

[10] HAJNÝ J. Anonymous Authentication for Smartcards. In: *Radioengineering.* Praha: Fakulta elektrotechnická ČVUT - Středisko vědeckotechnických informací, 2010, s. 363-368. 19, 2. ISSN 1210-2512.

[11] HAJNÝ J., MALINA L. *Unlinkable attribute-based credentials with practical revocation on smart-cards.* In: Smart Card Research and Advanced Applications. [online] Springer Berlin Heidelberg, 2013. s. 62-76. [cit. 2013-04-25]. Dostupné z: <`http://link.springer.com/content/pdf/10.1007%2F978-3-642-37288-9_5`>.

[12] NFCForum-TS-DigitalProtocol-1.0. *NFC Digital Protocol.* 2010.

[13] NFCForum-TS-NDEF_1.0. *NFC Data Exchange Format (NDEF).* 2006.

[14] NFCForum-TS-SNEP_1.0. *Simple NDEF Exchange Protocol.* 2011.

[15] ALPÁR G., HOEPMAN J., LUEKS W. *An Attack Against Fixed Value Discrete Logarithm Representations* [online]. 2013 [cit. 2013-05-29]. Dostupné z: <http://eprint.iacr.org/2013/120.pdf>. Cryptology ePrint Archive: Report 2013/120.

[16] HASELSTEINER E., BREITFUß, K. *Security in near field communication (NFC).* In: Workshop on RFID Security RFIDSec. [online] 2006. [cit. 2013-05-29]. Dostupné z: <http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002-%20Security%20in%20NFC.pdf>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

AES	Advanced Encryption Standard
APK	Android application package
ASK	Amplitude Shift Keying – amplitúdové kľúčovanie
AuC	Authentication Center – Autentifikačné centrum
DES	Data Encryption Standard
DL-REP	Discrete Logarithm Representation – reprezentácia diskrétneho logaritmu
DSA	Digital Signature Algorithm
EDGE	Enhanced Data Rates for GSM Evolution
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HLR	Home Location Register
HMAC	Hash-based Message Authentication Code
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
HTTP	Hypertext Transfer Protocol over SSL
ICCID	Integrated Circuit Card Identifier
IrDA	Infrared Data Association
IMSI	International Mobile Subscriber Identity
JAR	Java Archive
LAI	Location Area Identity
LTE	Long Term Evolution
MSC	Mobile Switching Center
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
REST	Representational State Transfer
RR	Revocation Referee

SHA-1	Secure Hash Algorithm
SIM	Subscriber Identity Module
SNEP	Simple NDEF Exchange Protocol
SRES	Signed Response
UMTS	Universal Mobile Telecommunications System
WiFi	Wireless Fidelity
XML	Extensible Markup Language
K_c	šifrovací kľúč
K_i	individuálny autentizačný kľúč
R_{E}	miera neúspešnosti identifikácie
$R_{\mathrm{FA}}(\Theta)$	pomerný počet chýb nesprávneho prijatia
$R_{\mathrm{FR}}(\Theta)$	pomerný počet chýb nesprávneho odmietnutia
R_{S}	miera úspešnosti identifikácie
Θ	verifikačný prah

ZOZNAM PRÍLOH

A AUTENTIZAČNÝ PROTOKOL HM12

Užívateľ **Čítačka**

$A_{seed}, g_1, g_2, g_3, n$

w_1, w_2, w_{RR}
$K_S \in_R \{0,1\}^{80}$
$A = A_{seed}^{K_S} \mod n$
$C_1 = g_3^{K_S w_{RR}}$
$C_2 = g_3^{K_S} \mod n$
$r_1 \in_R \{0,1\}^{400}$
$r_2 \in_R \{0,1\}^{400}$
$r_3 \in_R \{0,1\}^{600}$
$r_S \in_R \{0,1\}^{320}$
$\bar{A}_{seed} = g_1^{r_1} g_2^{r_2} g_3^{r_3} \mod n$
$\bar{A} = A_{seed}^{r_S} \mod n$
$\bar{C}_1 = g_3^{r_3} \mod n$
$\bar{C}_2 = g_3^{r_S} \mod n$
$e = \mathcal{H}(A, \bar{A}, A_{seed}, \bar{A}_{seed}, C_1, C_2, \bar{C}_1, \bar{C}_2, t)$
$z_1 = r_1 - eK_S w_1$
$z_2 = r_2 - eK_S w_2$
$z_3 = r_1 - eK_S w_{RR}$
$z_S = r_1 - eK_S$

$$A, C_1, C_2, e, z_1, z_2, z_3, z_S \longrightarrow$$

$\bar{A}_{seed} = A^e g_1^{z_1} g_2^{z_2} g_3^{z_3} \mod n$
$\bar{A} = A^e A_{seed}^{z_S} \mod n$
$\bar{C}_1 = C_1^e g_3^{z_3} \mod n$
$\bar{C}_2 = C_2^e g_3^{z_S} \mod n$
$e \stackrel{?}{=} \mathcal{H}(A, \bar{A}, A_{seed}, \bar{A}_{seed}, C_1, C_2, \bar{C}_1, \bar{C}_2, t)$

Obr. A.1: Autentizačný protokol HM12 [11] v detaile (`ProveAtt`)

$K_S \in_R \{0,1\}^l$ – náhodne generované číslo K_S o bitovej dĺžke l

$A_{seed}, g_1, g_2, g_3, n$	generačné konštanty
w_1, w_2, w_{RR}	tajné kľúče
K_S	tajný kľúč aktuálneho sedenia
e	Hash $\mathcal{H}$ funkcie SHA-1 s výstupom 160 bitov
z_1, z_2, z_3, z_S	odpoveď

B XML FORMÁT TOKENOV

```
<?xml version="1.0" encoding="UTF-8" ?>
<tokens>
  <!-- Generované konštanty -->
  <generated>
    <aseed></aseed>
    <g1></g1>
    <g2></g2>
    <g3></g3>
    <n></n>
  </generated>

  <!-- Tajné kľúče -->
  <secrets>
    <w1></w1>
    <w2></w2>
    <wrr></wrr>
  </secrets>
</tokens>
```

Pozn.: Všetky čísla sa uvádzajú v desiatkovej sústave.

C DVD PRÍLOHA

Popis obsahu priloženého DVD:

`beam-server/`	zdrojové kódy serverovej aplikácie *beam-server*
`drivers/`	ovládače NFC čítačky ACR122U
`NfcAuthZK/`	zdrojové kódy Android aplikácie *NfcAuthZK*
`OS/`	virtuálny operačný systém Ubuntu
`PrimitivesTest/`	zdrojové kódy Android aplikácie *EncryptionPrimitivesTest*
`packages/`	balíčky APK a JAR všetkých aplikácií
`tokens/`	tokeny autentizačného protokolu